Bayesian Biclustering on Discrete Data: Variable Selection Methods

Bayesian Biclustering on Discrete Data: Variable Selection Methods

A dissertation presented

by

Lei Guo

to

The Department of Statistics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Statistics

Harvard University

Cambridge, Massachusetts

September 2013

Dissertation Advisor: Jun S. Liu **Lei Guo**

Bayesian Biclustering on Discrete Data: Variable Selection Methods

Abstract

Biclustering is a technique for clustering rows and columns of a data matrix simultaneously. Over the past few years, we have seen its applications in biology-related fields, as well as in many data mining projects. As opposed to classical clustering methods, biclustering groups objects that are similar only on a subset of variables. Many biclustering algorithms on continuous data have emerged over the last decade. In this dissertation, we will focus on two Bayesian biclustering algorithms we developed for discrete data, more specifically categorical data and ordinal data.

The international HapMap project has made available the single-nucleotide polymorphism (SNP) data of thousands of individuals across the world. We analyzed the SNPs data with our biclustering algorithm for categorical data and described the similarities between human populations. In contrast to existing methods, our method can locate the SNPs that are specific to subpopulation groups. This can provide insight to the genome-wide association study (GWAS) by eliminating SNPs that are common to different ethic groups. We also identified a number of SNPs that are linked to disease, and this thesis describes their behavior in different subpopulations. The biclustering process can be used as a variable selection step prior to existing population inference procedures.

Contents

Acknowledgments

The successful completion of my PhD has been a memorable moment in my life with the best years of my life spent at the Harvard University; Department of Statistics, where I was able to work with some of the most brilliant minds in the world. During this journey, I am grateful to many individuals who have contributed to my educational, personal and professional development and who have truly helped me to present meaningful knowledge to the world through years of research.

I am forever grateful to my advisor, Prof. Jun S. Liu as this dissertation would not have been possible without his constant guidance, encouragement, unyielding support and patience. I also sincerely appreciate the help from the rest of my thesis committee, which includes Prof. Tirthankar Dasgupta and Prof. David P. Harrington.

In addition, I would like to thank other faculty and staff members of the Department of Statistics and the Harvard community, which include Edoardo Airoldi, Joseph K. Blitzstein, Betsey Cogswell, Steven Finch, S.C. Samuel Kou, James Matejek, Xiao-Li Meng, Luke W. Miratrix, Carl N. Morris, Alice Moses, Dale Rinkel, Donald B. Rubin, Maureen Stanton, and Darryl E. Zeigler.

For my entire PhD life I've been so fortunate to work with my amazing labmates, and I would like to thank: Ke Deng, Jiong Du, Daniel Fernandez, Simeng Han, Bo Jiang, Yang Li, Xuxin Liu, Yang Liu, Ping Ma, Di Wu, Jiexing Wu, Chao Ye for their enduring encouragement, camaraderie and the wonderful memories that they have provided for a lifetime.

Throughout this time, my parents have fully supported me and have given me endless help and encouragement. I am extremely blessed to have them in my life and proud to be their son.

This long journey would not have been the same without the support of my friends and special thanks go to: Rex G. Baker IV, Nickolas P. Chelyapov, Amanda Cheng, Chao Du, Linglan Gong, Jian Guo, Oliver Hayen, Konstantina Karterouli, Xiaoyan Peng, Melissa Rick, Lei Shen, Nan Shen, Yujun Wu, Ying Yan and Rie Yano.

To my beloved parents.

Chapter 1

Introduction

Clustering is the art of organizing similar objects into groups according to their variables so that objects are more similar to each other within each groups. It is a common technique in statistical data analysis, and has applications in many fields, *e.g.*, biological sequence analysis, population structure inference, medical imaging, market research, social networking analysis, recommender systems, search engine optimization, etc.

Central to the problem of clustering is how one defines a "cluster." The notion of a cluster can be defined in many ways, resulting in many different clustering algorithms (Estivill-Castro (2002)). Typical cluster models include *connectivity models*, which choose a measure of distance and then perform clustering based on distance connectivity; *centroid models*, which characterize clusters using a single centroid vector; *distribution models*, which employ statistical models to describe the clusters; *density models*, which define clusters as connected dense regions in the data space; *subspace models*, which select a subset of attributes and define clusters based on this subset of

space; etc. *Subspace models* are also called *biclustering*, or *two-way clustering*, which is the model this dissertation will explore in more detail in the following chapters.

Clustering algorithms produce different results based on their particular definitions of "cluster." Connectivity-based hierarchical clustering treats clusters as closely connective objects, and describes a cluster by the maximum distance to connect all parts of such cluster. There are also multiple choices for distance measure, *e.g.*, *single linkage*, *complete linkage*, *average linkage*, *ward*, *median*, etc. More clusters will form as distance increases, and this can be represented with a dendrogram, with the x-axis as the objects and the y-axis as the distance for the clusters to merge. There is no single partitioning of the data set but a hierarchy of clusters at different levels. *K-means* is another popular clustering algorithm based on *centroid models*. Given K (the number of clusters) the algorithm iteratively assigns each object into its nearest cluster and calculates the cluster's centroid. This is a NP-hard optimization problem and can usually be approximated. The clustering model most used by statisticians is distribution model-based clustering, or model-based clustering. The major benefit of model-based clustering is that the clusters are clearly defined as /textitobjects that share the same distribution. The interpretation of the clustering results is also straightforward, with each fitted parameter having its context based meanings. However, a known problem with model based clustering is overfitting. When we add more parameters into the model, we can always explain the data better, but the complexity of the model grows. Model complexity penalties are necessary for choosing the appropriate model.

Biclustering takes the task to the next level by seeking objects that are similar

over a subset of variables. The concept was first introduced by Hartigan (1972), and the term *biclustering* was coined by Mirkin (1996). However, for almost 30 years, the technique has seen no application in real data. In the year 2000, as more and more gene expression data was becoming available, Cheng and Church reintroduced the same concept and applied it to the gene expression data of yeast (Cheng and Church (2000)).

To further illustrate the concept, let us consider a rectangular matrix **M**, with **I** rows and **J** columns. Rows represent objects and columns represent features or variables. *Biclustering* algorithms seek to find a sub-matrix: a subset of rows that share similar patterns across a subset of columns. A simple example is consumers' purchase behavior with respect to clothing. Variables for clothes include color, style, material, texture, size, etc. Some people may only consider style, color, and texture when making a purchase, while other people may care about style and material. Based on the preferences of people, we can divide them into two separate groups, one using style, color and texture variables, the other one using style and material. Those form two biclusters and we used only a subset of the physical properties of clothes for each group. Another example for illustrating the concept of biclustering is the subject matter of documents. After removing commonly used words like **a**, **the**, **do**, etc., we will have a data matrix. Each row of the matrix represents a document and each column represents the counts of the occurrences of words that appeared in the document. We can thus find a subset of words and use those words to group similar documents together. Each of those groups are assumed to include documents with the same subject matter. For those problems, the data matrix contains many

variables but the groups are defined only using a subset of the variables. Traditional partitioning methods such as *k-means* will often produce undesirable results and are not ideal algorithms for classification.

Just as with traditional clustering, different definitions of biclusters inform different biclustering algorithms (Madeira and Oliveira (2004)). There are four major types of biclusters: **(a)** Biclusters with constant values; **(b)** Biclusters with constant values on rows (or columns); **(c)** Biclusters with coherent values (additive or multiplicative); and **(d)** Biclusters with coherent evolutions. Many biclustering algorithms have been developed since the application of biclustering to gene expression data, and we will review some notable algorithms in the next chapter.

This dissertation will focus on introducing two new biclustering algorithms, and one related application, to Human SNPs data from the HapMap project. Chapter 2 will give an overview of popular biclustering algorithms and their applications in different fields. Chapter 3 will introduce the basics of categorical data, drawing examples from different subjects, and will explain the theoretical background for Bayesian biclustering on categorical data. Chapter 4 will illustrate scenarios relating to the usage of ordinal data, and will also detail the Bayesian framework for biclustering on ordinal data using a Normal Random Cutoff approximation model. We also further extend the biclustering model with the capacity to handle more levels of ordinal data by introducing a Uniform Binomial mixture model. Chapter 5 will focus on the application of the categorical biclustering model to human single-nucleotide polymorphism (SNP) data from HapMap Phase III, as well as to disease linked SNPs analysis. Therein, we also present significant findings and result analysis.

Chapter 2

An overview of Biclustering Methods

2.1 Notation

We will now introduce a few notations to formally define *bicluster*. These notations will be used throughout the rest of the Chapter.

Suppose we have a data matrix **A**, with rows as object set X and columns as variable set Y and the entry a_{ij}. The purpose of *biclustering* methods is to find a sub-matrix of **A** with row set I $\subset$ X and column set J $\subset$ Y, such that the I objects are as similar as possible on column set J. This sub-matrix $A_{\{I,J\}}$ is called a *bicluster*, as seen in Figure 2.1. We use $a_{.j}$ to denote the mean of the jth column in the bicluster, $a_{i.}$ as the mean of the ith row in the bicluster, and $a_{..}$ as the overall mean of all elements of sub-matrix $A_{\{I,J\}}$.

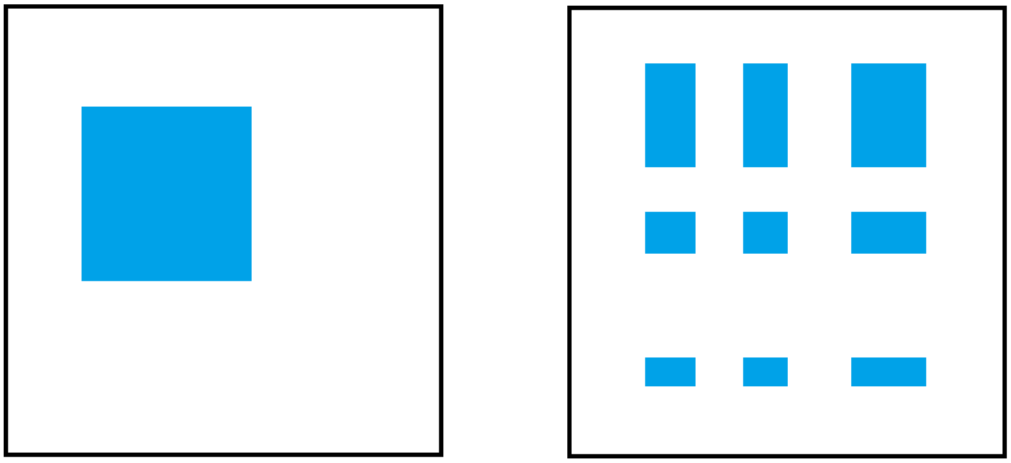

Figure 2.1: Illustration of a Bicluster. For a single bicluster, after re-arranging rows and columns, we can change it from the configuration on the right panel to the one on the left.

2.2 Types of Biclusters

By the nature of the data, there are two types of biclusters: (a) Biclusters with quantitative values and (b) Biclusters with qualitative values. According to the configuations of the biclusters they detect, Madeira and Oliveira (2004) further *clustered* existing biclustering algorithms into four major classes:

1. Biclusters with constant values.

   constant values on rows and columns

   $$a_{ij} = \mu$$

2. Biclusters with constant values on columns or rows.

   constant values on rows $a_{ij} = \mu + \alpha_i$

   constant values on columns $a_{ij} = \mu + \beta_j$

3. Biclusters with coherent values.

   additive $a_{ij} = \mu + \alpha_i + \beta_j$

   multiplicative $a_{ij} = \mu \cdot \alpha_i \cdot \beta_j$

4. Biclusters with coherent evolutions.

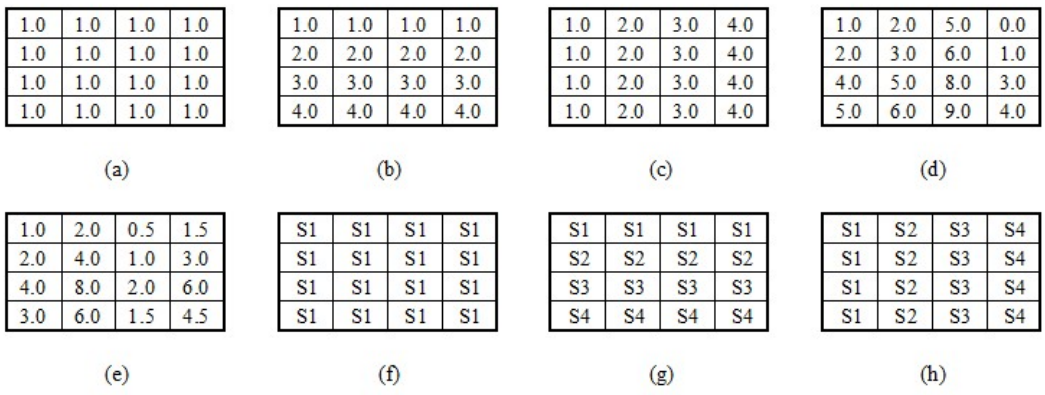

Figure 2.2: Illustration of the four major types of biclusters that existing algorithms seek to recover.

Figure 2.2 displays all four types of different biclusters. In Figure 2.2a, values are all the same for each cell in the data matrix. In Figure 2.2b, every row has identical

values while different rows have different values. In Figure 2.2c, every column has identical values while different columns have different values. Figure 2.2b and Figure 2.2c are equivalent if we transpose the data matrix. In Figure 2.2d, every cell is a summation of row effect and column effect, each row has a different row effect, and similarly for different columns. In Figure 2.2e, every cell is a multiplication of row effect and column effect, each row has a different row effect, and similarly for different columns. Figure 2.2d and Figure 2.2e are essentially the same if we take the logarithm of the data matrix. The first three types of biclustering algorithms deal with the numeric values of the data matrix directly. They aim to find subsets of rows that are similar over corresponding subsets of columns. Because these three types of biclustering use the numeric values of a data matrix directly, many related biclustering algorithms have been developed, which we will illustrate in the following sections.

The fourth type of biclustering algorithm deals with coherent evolutions, regardless of the real numeric values in the data matrix. The biclustering is performed on an abstract layer of the data. This abstract layer views data as symbols. There are two types of symbols: non-ordered symbols, that is, nominal (categorical); and order-preserving symbols, that is, ordinal. Few algorithms have been developed for coherent evolution data and most of them are simply ad-hoc. This dissertation presents a general framework for the biclustering of these two types of data, and will present the details thereof in Chapters 3 and 4. Our biclustering algorithm for categorical and ordinal data can handle both numeric matrices and matrices with coherent evolution data. The continuous case for coherent value (additive and multiplicative) bicluster-

ing has been covered by Gu and Liu (2008).

2.3 Patterns of Biclusters

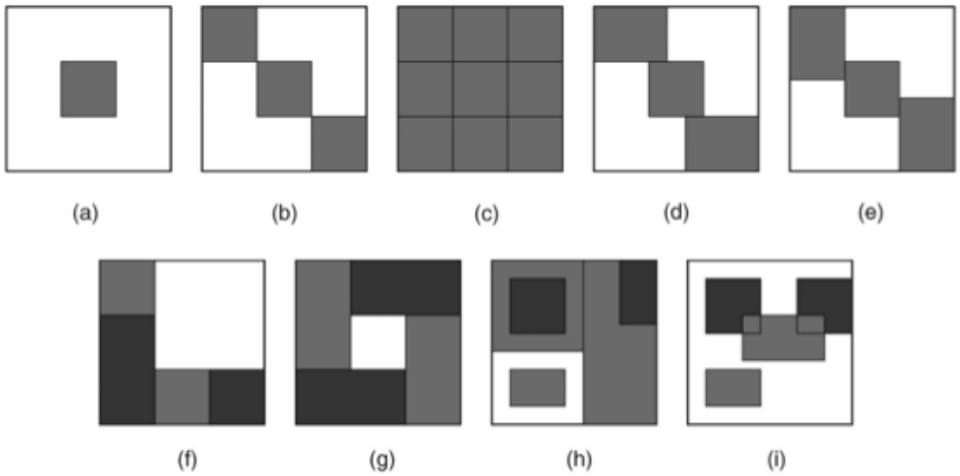

Figure 2.3: Different structures of biclusters, from Madeira and Oliveira (2004)

There are generally more than one bicluster in a data matrix. Let's assume there are **K** biclusters in a data matrix. In the ideal case, after reordering rows and columns of the data matrix, the biclusters may appear as rectangular blocks. Within each of those blocks, the objects are more similar to each other in terms of their values on the chosen subset of columns.

For most algorithms, we assume a neutral background where little information can be captured for the elements outside the bicluster blocks. However, in our model, we do not assume the existence of a background, and allow columns that are shared across biclusters to be *background* in the traditional meaning. These *background* columns can be eliminated for variable selection purposes.

Depending on the relative positioning of those blocks in the data matrix, Madeira and Oliveira (2004) stated there are eight different configurations for the bicluster structures after reordering rows and columns, as plotted in 2.3.

1. Biclusters with exclusive rows and columns.
2. Nonoverlapping biclusters with checkerboard structure.
3. Exclusive-rows biclusters.
4. Exclusive-columns biclusters.
5. Nonoverlapping biclusters with tree structure.
6. Nonoverlapping nonexclusive biclusters.
7. Overlapping biclusters with hierarchical structure.
8. Arbitrarily positioned overlapping biclusters.

This is an exhaustive list of possible relative positioning of biclusterings. However, from a variable selection perspective, we can unify them into one single configuration: biclusters with exclusive rows. First, biclusters with exclusive columns are equivalent to biclustering with exclusive rows by transposing the data matrix. Biclusters with exclusive rows and columns can be viewed as a special case of biclusters with exclusive rows. Non-overlapping biclusters with the checkerboard structure in Figure 2.3c can be further grouped into three larger biclusters each with exclusive rows. Non-overlapping biclusters with the tree structure in Figure 2.3f can be grouped into three new clusters with cluster 1 and cluster 2 only having the first one-thirds of the

column sets. Similarly, for Non-overlapping nonexclusive biclusters in Figure 2.3g, three new biclusters can be formed with cluster 2 having a gap in the middle of the bicluster. Overlapping biclusters with a hierarchical structure as displayed in Figure 2.3h are more complicated and can be dissected into seven new biclusters, each with exclusive rows (some of them have gaps in them). Following the same logic, we can separate the arbitrarily positioned overlapping biclusters in Figure 2.3i into several smaller biclusters with exclusive rows. Under our model setting, we can simplify all the above bicluster patterns to a single chessboard structure, and everything is a special case for this structure. Two examples are presented in Figure 2.4 and Figure 2.5.

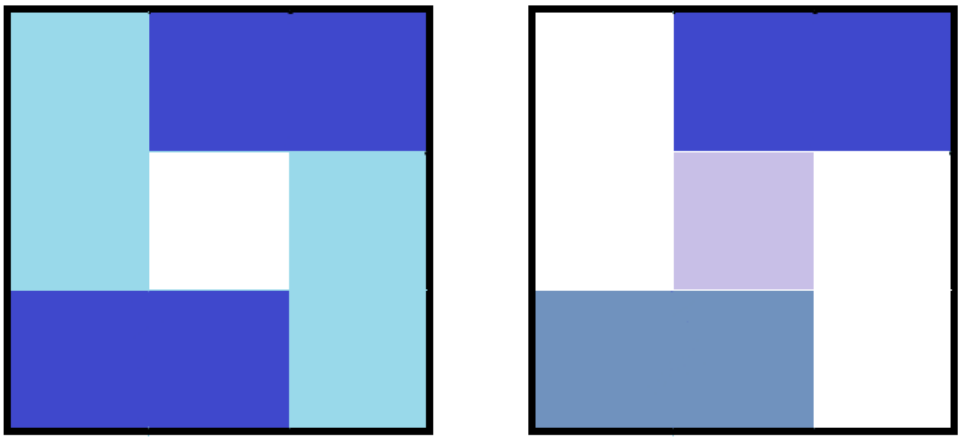

Figure 2.4: Non-overlapping Non-exclusive biclusters mapped to three row-exclusive, column-non-exclusive biclusters under our definition of biclusters.

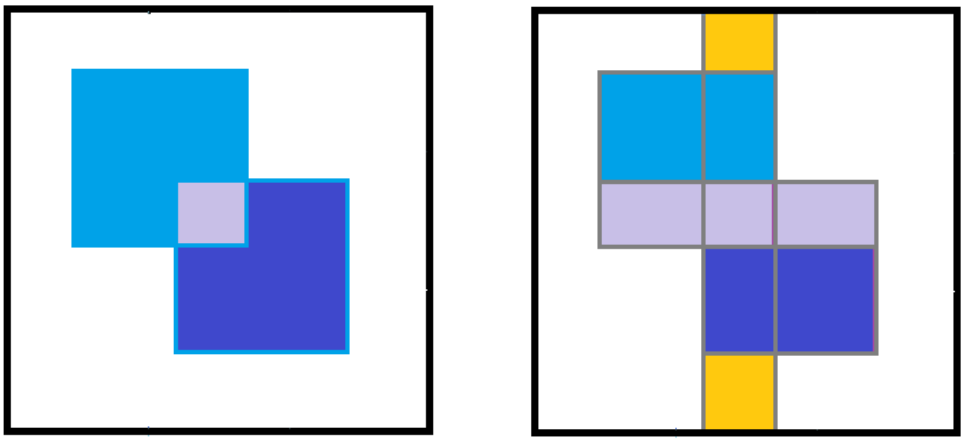

Figure 2.5: Two overlapping biclusters mapped to five biclusters in BBCD.

The essential part of this concept is to assign every row into a bicluster. If every column of the chosen bicluster is the same as the corresponding column across all other biclusters, this bicluster will be treated as trivial, and be discarded. This idea will be further illustrated in Chapter 3 as we explain our biclustering model on categorical data.

2.4 Biclustering algorithms

1. **Block Clustering**

   The first biclustering algorithm ever developed was by Hartigan (1972), which is also called block clustering. Block clustering aims to find biclusters with constant values across the data matrix. Supposing there are K biclusters in the data, an ideal case would be that within each of those K biclusters the entry values a_{ij} are identical. They use a SSQ (Sum of Squares) to measure the deviation of the found biclusters from the ideal model.

   Given a particular partition B_1, B_2, ..., B_K, where B_k is the sets of rows and columns of bicluster k in the data matrix, the deviation is defined as:

$$SSQ = \sum_{k=1}^{K} \sum_{i,j \in B_k} (a_{ij} - b_k)^2$$

   where b_k is the average of all entries in partition B_k, or in other words, in bicluster k.

   The algorithm starts with a partition that consists of the full data set. It proceeds by splitting selected partitions. At the kth step, a partition B_p is chosen for splitting and then the partition will change from B_1, B_2, ..., B_p, ..., B_k to B_1, B_2, ..., B_{p-1}, $B_p^{'}$, $B_p^{''}$ ..., B_k, which increases the total number of biclusters by 1. The split can happen either on rows or on columns. The algorithm selects the splitting that will maximize the SSQ reduction at the kth step, and stops when the total number of biclusters reaches K.

   This is the first biclustering algorithm, which built the foundation for later

biclustering works.

2. **Bayesian Biclustering for Continuous data (BBC)**

   Gu and Liu (2008) developed a Bayesian Biclustering model for continuous data and used a Gibbs sampling procedure to infer the bicluster structure in the data matrix. They introduced two normalization methods for data processing: the interquartile range normalization and the smallest quartile range normalization. Similar to the Plaid model (Lazzeroni and Owen (2002)), in BBC, gene expression value a_{ij} in a bicluster k is assumed to be the summation of the additive effects of cluster specific background level μ_k, gene effect α_{ik}, condition effect β_{jk}, and a noise term ϵ_{ijk}. Entries that do not belong to any cluster are described by a noise term ϵ_{ij}.

$$A_{ij} = \sum_{k=1}^{K}[(\mu_k + \alpha_{ik} + \beta_{jk} + \epsilon_{ijk}) \cdot \rho_{ik}\kappa_{jk}] + \epsilon_{ij}(1 - \sum_{k=1}^{K} \rho_{ik}\kappa_{jk})$$

   where $\rho_{ik} \in \{0, 1\}$ is the gene bicluster membership indicator; $\kappa_{jk} \in \{0, 1\}$ is the condition bicluster membership indicator; and A_{ij} is in bicluster k if and only if $\rho_{ik} = \kappa_{jk} = 1$.

   Unlike the ordinal Plaid model, BBC only allows biclusters to overlap in either row direction or column direction, which results in two versions of BBC: non-overlapping gene biclustering and non-overlapping condition biclustering. In non-overlapping gene biclustering, a gene can be assigned into at most one cluster, while a condition can be assigned into multiple biclusters. The constraints can be represented as $\sum_{k=1}^{K} \rho_{ik} \leq 1$. In either version of BBC, there

are no overlapping elements between different clusters.

The priors for the membership indicators are set as:

$$\kappa_{jk} \sim Bernoulli(q_k)$$

$$P(\rho_{ik} = 1, \rho_{il} = 0, l \neq k) = p_k$$

$$P(\rho_{il} = 0, l = 1, 2, \ldots, K) = p_0 = 1 - \textstyle\sum_{k=1}^{K} p_k$$

Initial values for q_k are set to be 0.5 and $p_k = \frac{1}{K+1}$. Different choices of initial values of q_k and p_k do not affect the results much.

Priors for other parameters are set as:

$$\begin{aligned}
\mu_k &\sim N(0, \sigma^2_{\mu_k}) \\
\alpha_{ik} \mid \rho_{ik} = 1 &\sim N(0, \sigma^2_{\alpha_k}) \\
\beta_{jk} \mid \kappa_{jk} = 1 &\sim N(0, \sigma^2_{\beta_k}) \\
\epsilon_{ijk} &\sim N(0, \sigma^2_{\varepsilon_k}) \\
\epsilon_{ij} &\sim N(0, \sigma^2_{\varepsilon})
\end{aligned}$$

Hyperpriors for $\sigma^2_{\mu_k}, \sigma^2_{\alpha_k}, \sigma^2_{\beta_k}, \sigma^2_{\varepsilon_k}, \sigma^2_{\varepsilon}$ are all from Inverse Gamma distributions. Denote all hyperpriors as a $\boldsymbol{\sigma}$ vector.

Under this setting, the probability distribution of a_{ij} can be written as:

$$a_{ij} \sim \begin{cases} N(\mu_k + \alpha_{ik} + \beta_{jk}, \sigma^2_{\varepsilon_k}) & \text{if } \rho_{ik} \cdot \kappa_{jk} = 1; \\ N(0, \sigma^2_{\varepsilon}) & \text{if } \rho_{ik} \cdot \kappa_{jk} = 0. \end{cases}$$

The conditional marginal distribution of a_{ij} is:

$$Y \mid \rho, \kappa \sim N(0, \Sigma)$$

where $\mathbf{Y} = (Y_0, Y_1, \ldots, Y_K)^T$ with $Y_k = \{a_{ij} : \rho_{ik}\kappa_{jk} = 1\}, k \geq 1$; Σ is the covariance matrix of $\mathbf{Y}$.

The membership indicator ρ and κ can be updated iteratively according to:

$$\frac{P(\kappa_{jk} = 1 \mid \kappa_{[-jk]}, \rho, \sigma, Y)}{P(\kappa_{jk} = 0 \mid \kappa_{[-jk]}, \rho, \sigma, Y)}$$

$$\frac{P(\rho_{ik} = 1 \mid \rho_{[-ik]}, \kappa, \sigma, Y)}{P(\rho_{ik} = 0 \mid \rho_{[-ik]}, \kappa, \sigma, Y)}$$

With the membership indicators, one can recover the bicluster structures in the data matrix. The number of biclusters K can be determined by running the algorithm with different values of K and selecting one according to the Bayesian Information Criterion (BIC) (Schwarz (1978)).

3. **Cheng and Church δ-biclustering**

Cheng and Church (2000) proposed a δ-biclustering method for the gene expression data. Their definition of bicluster is the same as the additive Plaid model (Lazzeroni and Owen (2002)): each entry in a bicluster can be viewed as a summation of a constant cluster-specific background level, gene effect and condition effect. After those effects are removed, the residual levels should be as small as possible, measured using a pre-defined threshold δ. Following our

notation at the beginning of this chapter, given the gene expression matrix A, genes subset I and conditions subset J, we define:

$$\begin{aligned} a_{.j} &= \frac{\sum_{i \in I} a_{ij}}{\mid I \mid} \\ a_{i.} &= \frac{\sum_{j \in J} a_{ij}}{\mid J \mid} \\ a_{..} &= \frac{\sum_{i \in I, j \in J} a_{ij}}{\mid I \mid\mid J \mid} \end{aligned}$$

For entry a_{ij} in the bicluster, its residual can be calculated as $r_{ij} = a_{ij} - a_{i.} - a_{.j} + a_{..}$. The mean square residual score of the submatrix $A_{\{I,J\}}$ is

$$H(I, J) = \sum_{i \in I, j \in J} \frac{r_{ij}^2}{\mid I \mid\mid J \mid}$$

The goal is to find a locally maximal sub-matrix with a score smaller than δ. There are two phases in the algorithm: decay and growth. It starts with the full data matrix as the desired bicluster, for each row it calculates the average residual as $r_i = \frac{1}{|J|} \sum_{j \in J} r_{ij}$, and for each column it calculates $r_j = \frac{1}{|I|} \sum_{i \in I} r_{ij}$. The row or column with the highest average residual value will be removed from the bicluster. This iterates until $H(I, J)$ is below threshold δ. In the second phase of the algorithm, it seeks to add rows or columns with the lowest average residual values to the bicluster, under the constraint that $H(I, J) < \delta$.

This algorithm can detect biclusters one at a time. To find more biclusters, the identified bicluster blocks have to be replaced with random noise to prevent it from being included in the new bicluster.

4. **Coupled Two-way Clustering**

   Getz et al. (2000) introduced an algorithm for gene microarray data analysis called Coupled Two-way Clustering (CTWC). The algorithm uses a traditional one-way clustering method to find clusters on rows and columns iteratively. To do this, a *stationary cluster* is defined as a genes subset V' and conditions subset U' in a larger genes set V and conditions set U, such that when traditional clustering is performed on V, the columns of the *stationary cluster* V' can be recovered as a significant cluster. Similarly, the genes set V' can also be recovered by performing one-way clustering on the larger genes set V.

   The algorithm starts with the whole data matrix and iteratively select a genes subset V and conditions subset U. Traditional one-way clustering is then applied to the sub-matrix $V \times U$. If a *stationary cluster* is found, then the rows and columns of the *stationary cluster* will be added to the respective selected genes and conditions set. The algorithm proceeds until no new *stationary cluster* can be found. The performance of Coupled Two-way Clustering also depends on the traditional one-way clustering algorithm employed. Some algorithms that cannot distinguish significant clusters from non-significant ones cannot be embedded into the Coupled Two-way Clustering.

5. **The Iterative Signature Algorithm**

   Bergmann et al. (2003) proposed a biclustering method specially designed for noisy gene expression data known as ISA (*the Iterative Signature Algorithm*). In their algorithm, they define bicluster as a gene transcription module such that the expression levels for the genes in the bicluster are significantly higher

over every chosen condition in the bicluster, which can be measured using a Z-score. The algorithm iteratively searches for the sets of genes and conditions until the desired bicluster is found.

ISA aims to find a special bicluster such that the conditions of a bicluster uniquely determine the objects and vice versa. In a mathematic framework, if we standardize row-wise and column-wise a data matrix to generate E^G and E^C respectively, a bicluster $B = (U', V')$ is defined as the combination of U' and V' which satisfies

$$U' = \{u \in U : |e^C_{uV'}| > T_C\sigma_C\}, V' = \{v \in V : |e^G_{U'v}| > T_G\sigma_G\}$$

simutaneously. The definition is intuitively reasonable under normal assumption.

The algorithm is formalized as follows. We start with a set of individuals V_0 arbitrarily or based on some prior information. U' and V' can be iteratively updated by

$$U_i = \{u \in U : |e^C_{uV_i}| > T_C\sigma_C\}, V_{i+1} = \{v \in V : |e^G_{U_iv}| > T_G\sigma_G\}$$

The algorithm terminates at step n such that

$$\frac{|V_{n-i} \setminus V_{n-i-1}|}{|V_{n-i} \cup V_{n-i-1}|} < \epsilon$$

In the algorithm, T_C, T_G, σ_G, σ_G and ϵ are all pre-specified and are tunable. Dif-

ferent settings of these parameters and V_0 can be tried to detect a representative set of biclusters.

6. **Plaid Model**

Lazzeroni and Owen (2002) developed a statistical algorithm called the Plaid model to deal with biclusters with additive or multiplicative coherent values. Multiplicative coherent values can be converted into additive values by taking the logarithm of the original data matrix. Here we will illustrate the additive Plaid model. The basic concept is to treat the data matrix as a superposition of layers of data. Each layer is a bicluster, with each entry as a cluster specific background level plus a row effect and a column effect. Under this setting, the Plaid model can deal with overlapping biclusters.

Think of the data matrix as a gene expression data set, with rows as genes and columns as conditions, the expression matrix can be represented as

$$A_{ij} = \mu_0 + \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$$

where μ_0 is the overall background level, $\rho_{ik} \in \{0, 1\}$ is the gene membership indicator for the bicluster, $\kappa_{jk} \in \{0, 1\}$ is the condition membership indicator for the bicluster; $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$, where μ_k is the bicluster specific background level for bicluster k, α_{ik} is the additive gene effect for the i^{th} gene, β_{jk} is the additive condition effect for the j^{th} condition.

Finding the biclusters then boils down to the following minimization problem:

$$\sum_{i,j}(A_{ij} - \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk})^2$$

where $\theta_{ij0} = \mu_0$, $\rho_{i0} = \kappa_{j0} = 1$. To make the model identifiable, constraints $\sum_i \rho_{ik}^2 \alpha ik = 0$ and $\sum_j \kappa_{jk}^2 \beta_{jk} = 0$ are imposed. The layers are added one at a time and at each layer-finding step we choose the layer that minimizes the sum of squared errors.

Suppose we have $K-1$ layers, to seek for the K^{th} layer, we want to minimize

$$\begin{aligned} Q^{(K)} &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(Z_{ij}^{(K-1)} - \theta_{ijK}\rho_{iK}\kappa_{jK})^2 \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(Z_{ij}^{(K-1)} - (\mu_K + \alpha_{iK} + \beta_{jK})\rho_{iK}\kappa_{jK})^2 \end{aligned}$$

subject to $\sum_i \rho_{iK}^2 \alpha_{iK} = 0$ and $\sum_j \kappa_{jK}^2 \beta_{jK} = 0$

where

$$Z_{ij}^{(K-1)} = A_{ij} - \sum_{k=0}^{K-1} \theta_{ijk}\rho_{ik}\kappa_{jk}$$

is the sum of squared errors after removing the first $K-1$ layers.

Updating θ_{ijK}

The parameters for θ_{ijK} can be calculated using Lagrange multipliers:

$$\begin{aligned}
\mu_K &= \frac{\sum_i \sum_j \rho_{iK}\kappa_{jK} Z_{ij}^{(K-1)}}{(\sum_i \rho_{iK}^2)(\sum_j \kappa_{jK}^2)} \\
\alpha_{iK} &= \frac{\sum_j (Z_{ij}^{(K-1)} - \mu_K \rho_{iK}\kappa_{jK})\kappa_{jK}}{\rho_{iK} \sum_j \kappa_{jK}^2} \\
\beta_{jK} &= \frac{\sum_j (Z_{ij}^{(K-1)} - \mu_K \rho_{iK}\kappa_{jK})\rho_{iK}}{\kappa_{jK} \sum_i \rho_{iK}^2}
\end{aligned}$$

Updating ρ_{iK} and κ_{jK}

Given θ_{ijK}, ρ_{iK} and κ_{jK} that minimize $Q^{(K)}$ can be obtained as:

$$\begin{aligned}
\rho_{iK} &= \frac{\sum_j \theta_{ijK}\kappa_{jK} Z_{ij}^{(K-1)}}{\sum_j \theta_{ijK}^2 \kappa_{jK}^2} \\
\kappa_{jK} &= \frac{\sum_i \theta_{ijK}\rho_{iK} Z_{ij}^{(K-1)}}{\sum_i \theta_{ijK}^2 \rho_{iK}^2}
\end{aligned}$$

The parameters were updated iteratively for many steps. The layer K will only be accepted if the residual matrix $Z_{ij}^{(K-1)}$ contains non noise. Otherwise, the algorithm will stop and report $K-1$ as the total number of biclusters in the data. A permutation test is conducted to judge whether $Z_{ij}^{(K-1)}$ still has a certain pattern.

7. **Spectral Biclustering**

Spectral Biclustering was developed using a linear algebra method of Singular Value Decomposition (SVD) of a data matrix. Kluger et al. (2003) presented this method and showed that it applies to a gene expression matrix that has a hidden checkerboard-like structure.

Suppose we have a gene expression data matrix **E**, which has a hidden checkerboard structure. If we supply a vector **x** that matches the block pattern of the rows of **E**, we will get a vector **y** that reveals the column block structure of **E**. In other words, we can project the row block pattern of matrix **E** by multiplying it with a matching **x**. If we multiply **y** with E^T, we will get another vector **x'**, which has the same block patter as **x**. We can see that the block pattern of **x** forms a closed space under $E^T \cdot E$, which can be described as linear combination of the eigenvectors of matrix $E^T E$. Similarly, the eigenvectors of EE^T span the closed space formed by vectors with the block pattern of **y**, which is the block pattern of the columns of **E**.

$$E \cdot x = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 3 \\ 1 & 1 & 2 & 2 & 3 & 3 \\ 4 & 4 & 5 & 5 & 6 & 6 \\ 4 & 4 & 5 & 5 & 6 & 6 \end{pmatrix} \cdot \begin{pmatrix} a \\ a \\ b \\ b \\ c \\ c \end{pmatrix} = \begin{pmatrix} d \\ d \\ e \\ e \end{pmatrix} = y$$

$$E^T \cdot y = \begin{pmatrix} 1 & 1 & 4 & 4 \\ 1 & 1 & 4 & 4 \\ 2 & 2 & 5 & 5 \\ 2 & 2 & 5 & 5 \\ 3 & 3 & 6 & 6 \\ 3 & 3 & 6 & 6 \end{pmatrix} \cdot \begin{pmatrix} d \\ d \\ e \\ e \end{pmatrix} = \begin{pmatrix} a' \\ a' \\ b' \\ b' \\ c' \\ c' \end{pmatrix} = x'$$

$$E^T \cdot E \cdot x = E^T \cdot E \cdot \begin{pmatrix} a \\ a \\ b \\ b \\ c \\ c \end{pmatrix} = \begin{pmatrix} a' \\ a' \\ b' \\ b' \\ c' \\ c' \end{pmatrix} = x'$$

The eigenvectors and eigenvalues for $E^T E$ and EE^T can be obtained by performing a singular value decomposition on matrix **E**.

$$E = U \Sigma V^T$$

The columns of **U** and **V** are the corresponding eigenvectors for EE^T and $E^T E$ and the square of the diagonal elements are the corresponding eigenvalues shared by the eigenvector pairs. One can check the block pattern of each of the eigenvector pairs and find the corresponding biclusters.

8. **The SAMBA algorithm**

The Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) was developed by Tanay (Tanay et al. (2002), Tanay et al. (2005)), and converts the data matrix into a bipartite graph. For gene expression data, the two parts of the corresponding bipartite graph are genes and conditions, with edges for significant expression level changes. Let $G = (U,\ V,\ E)$ be the bipartite graph converted from the input expression data. V is the set of genes and U is the set of conditions. A vertex pair $(u,\ v) \in E$ if and only if there is a significant change of expression level for gene v under experimental condition u. Let $H = (U',\ V',\ E')$ be a subgraph of G, and let $\bar{E}' = (U' \times V') \setminus E'$. The null model assumes the occurrence of each edge $(u,\ v)$ is from a Bernoulli distribution with parameter $p_{u,v}$, while the alternative model assumes that each edge of a bicluster is from a Beroulli distribution with a constant p_c ($p_c > p_{u,v}$). The log likelihood for H can be written as:

$$log(L(H)) = \sum_{(u,v)\in E'} log\frac{p_c}{p_{u,v}} + \sum_{(u,v)\in \bar{E}'} log\frac{1-p_c}{1-p_{u,v}}$$

This log likelihood is used as a score function for the subgraph H. Finding the most significant bicluster in the data matrix now becomes a problem of finding the heaviest subgraphs in the converted bipartitie graph using the weight defined above for the subgraph. The algorithm then searches for subgraphs using a heuristic method by starting with heavy bicliques as seeds around each vertex of the graph. It iteratively modifies the current bicluster by adding or removing a vertex until no score improvement can be achieved.

9. **xMOTIF**

Murali and Kasif (2003) presented a biclustering method that defined bicluster as a group of genes that are simultaneously conserved across a subset of conditions. The found bicluster is called a gene expression motif (xMOTIF). The conservation of a gene can be quantified by asserting whether its expression values are in the same state across conditions. A gene state is a range of expression values that are statistically significant. A state is more interesting if it contains more expression values than one would expect if the values were generated at random. The null hypothesis is that the expression values of a gene are generated from a uniform distribution. Let interval [a, b] be the state we are interested in, and K out of the total of n values fall into this interval, we can compute the p-value of this state as:

$$\sum_{k \leq i \leq n} (b-a)^i (1-(b-a))^{(n-i)}$$

Here, a and b are both numbers between 0 and 1, because gene expression values lie in the interval [0, 1].

The states were chosen according to the p-values of the intervals. The algorithm starts from different conditions as seeds and tries to find the largest xMOTIF by adding gene-states that are most distinguishing for genes and the corresponding conditions. The found xMOTIF must satisfy the following: the number of conditions chosen is at least an α-fraction of all the conditions; for genes not in the xMOTIF, it is conserved in at most a β-fraction of the conditions; and,

every gene in the xMOFIT is conserved across all the chosen conditions, `e.g.`, in the same state.

The major logic behind xMOTIF is to extract from the data matrix an abstract layer: *states* and then use the *states* to perform the biclustering. It can handle coherent evolutions data with constant nominal patterns on rows or columns.

10. **Other Biclustering algorithms**

Many other biclustering methods have been published so far, applying to a variety of data types. Among them are FLOC (Yang et al. (2002) Yang et al. (2003)), pClusters (Wang et al. (2002)), PRMs (Segal et al. (2001)) and OPSMs (Ben-Dor et al. (2002)) etc.

Chapter 3

Bayesian Biclustering on Categorical Data

3.1 Background

Many biclustering algorithms have been developed since Cheng and Church (2000) applied their δ-biclustering to gene expression data. However, most of those algorithms deal with continuous data. Very few of existing methods are designed for discrete data. In this chapter, we propose a Bayesian Biclustering method for categorical (nominal) data. In Chapter 4 we will introduce another algorithm we developed under the Bayesian frame work for ordinal data.

3.2 Categorical Data

Categorical data, a type of discrete data sometimes called nominal data, is a statistical data type whose value is one of a number of fixed categories. There is no intrinsic ordering to these categories. One simple example for categorical data is the color of people's hair, which might be **black**, **red**, **brown**, **blond**, **brunette**, etc. There is no way to rank hair color from low to high. For categorical data, because there is no ordering, calculating the arithmetic mean does not make any sense. A distance based clustering method is no longer applicable in this situation. In this chapter we will mainly discuss biclustering for categorical data. We will leave the modeling of the ordinal data (the other kind of discrete data) for the next chapter.

The data we are interested in is an I by J rectangular data matrix Y, with rows representing objects and columns representing variables. There are **M** different categories for each entry a_{ij} in the data set, $y_{ij} \in \{1, 2, \ldots, M\}$. In reality, different variables may have different numbers of categories. We can easily extend our algorithm to be applicable to this scenario by allowing M to be variable specific, *e.g.*, $y_{ij} \in \{1, 2, \ldots, M_j\}$.

3.3 Modeling

Similar to existing biclustering methods, the primary goal of our biclustering algorithm is to find a subset of rows and columns such that the selected objects are more similar to each other over the selected columns. In addition, we are also interested in identifying those variables that are determinant of the biclusters, namely the

variables that distribute unevenly across all biclusters such that they can be used to characterize the biclusters. In our model, we assume that the objects are independent conditional on their bicluster assignment, and that there is no interaction between column variables.

Unlike existing approaches, we do not distinguish background and foreground from each other explicitly. We assign each object into one of the predefined biclusters, and each object can only be assigned into one bicluster. We include all columns in the bicluster, and use a pattern indicator for each column to describe the similarity pattern between biclusters. The majority of the biclusters that share the same distribution are assigned to the background of that column. This is a general definition and it can deal with all the previously listed bicluster structure patterns. Another advantage of our algorithm is that it can handle not only numerical biclusters with constant row or column values, but also biclusters with coherent evolutions, *e.g.*, *sign changes*, *nominal patterns*, etc.

3.3.1 Notations

Let K be the number of clusters in our algorithm, let Z_i represent the cluster ID for object i, and let $\mathbf{S_j}$ be the column pattern indicator for the j_{th} column, $j = 1, 2, \ldots, J$. Z_i can take values from $\{1, 2, \cdots, K\}$, $i = 1, 2, \ldots, I$. As mentioned above, each object will be and can only be assigned into one cluster.

As the pattern indicator for K clusters, S is a vector of length K and each of its elements can take a value of either 0 or 1. There are initially 2^K different configura-

tions.

$$\binom{K}{0}, \binom{K}{1}, \binom{K}{2}, \ldots, \binom{K}{K-1}, \binom{K}{K}$$

In describing the similarity pattern in our model,

$$\binom{K}{K-1}, \binom{K}{K}$$

are essentially equivalent. We remove this redundancy and use only all-1 vector $(1, 1, \ldots, 1)$ to refer to this specfic pattern. The final configurations of S reduces to $2^K - K$.

3.3.2 Model Settings

In our model, we assume that columns are independent of each other, rows are independent conditional on their cluster assignment. Given the bicluster assignment $Z_i = k$, each y_{ij} is from a multinomial distribution with frequency parameter $\vec{\theta}_{kj}$. Let Θ be the set of all $\vec{\theta}_{kj}$, which has 3 dimensions: K by J by M.

The distribution of the j^{th} variable of object i, given its cluster ID Z_i, distinction pattern indicator S_j, and frequency parameter Θ, can be expressed as follows:

$$y_{ij} \mid \mathbf{Z}, \mathbf{S}, \Theta \sim Multinom(\vec{\theta}_{Z_i,j})$$

The full likelihood of this model can be written as follows:

$$P(Y \mid \mathbf{Z}, \mathbf{S}, \Theta) = \prod_{i=1}^{I} \prod_{j=1}^{J} \theta_{Z_i,j,y_{ij}} \tag{3.1}$$

Priors

We model the assignment of Z_i as a Chinese Restaurant Process, and give the prior as:

$$P(\mathbf{Z}) \propto \frac{\Gamma(\alpha_z){\alpha_z}^K}{\Gamma(\alpha_z + I)} \prod_k \Gamma(C_k) \tag{3.2}$$

where C_k is the size of cluster k, α_z is the *concentration* parameter in Chinese Restaurant Process. Here we set $\alpha_z = 1$.

The priors for $\mathbf{S}$ was given to penalize the inclusion of distinctive clusters over columns.

We let

$$P(S \mid Z) \propto \prod_j \frac{a^{\sum_k S_{kj}}}{\sum_{S_j} a^{\sum_k S_{kj}}} \tag{3.3}$$

where $a < 1$ is a positive number. In our simulation study and real data application, we use $a = 0.05$.

We assume that the multinomial parameters for all column cluster specific categorical distributions are from the same Dirichlet distribution and give each $\vec{\theta}_{k,j}$ a Dirichlet prior as:

$$\vec{\theta}_{k,j} \mid Z, S \sim Dirichlet(\vec{\alpha}_\theta); \tag{3.4}$$

$$P(\Theta \mid Z, S) \propto \prod_j \left(\prod_{k:S_{kj}=1} \prod_{m=1}^{M} \theta_{k,j,m}^{\alpha_\theta - 1} \cdot \prod_{k:S_{kj}=0} \prod_{m=1}^{M} \theta_{k,j,m}^{\alpha_\theta - 1} \right) \tag{3.5}$$

In column j, for the clusters that share the same θ, the second term in the bracket is multiplied only once. Here we set common values for each category across all

clusters in $\vec{\alpha}_\theta$ as:

$$\vec{\alpha}_\theta \;=\; \{1, 1, \cdots, 1\}$$

When the number of categories in the data is large, it is reasonable to use a fixed total pseudo-counts and allocate them equally to each dimension of $\vec{\alpha}_\theta$. In our simulation and application, there are only 3 categories and we used $\{1, 1, 1\}$.

Under above setting, the joint posterior of the model can be written as

$$\begin{aligned} P(Z, S, \Theta \mid Y) &\propto P(Y \mid Z, S, \Theta) \cdot P(Z) \cdot P(S \mid Z) \cdot P(\Theta \mid Z, S) \\ &\propto \prod_i \prod_j \theta_{j, S_j, Z_i, y_{ij}} \\ &\cdot \frac{\Gamma(\alpha_z) {\alpha_z}^K}{\Gamma(\alpha_z + I)} \prod_k \Gamma(C_k) \\ &\cdot \prod_j \frac{a^{\sum_k S_{kj}}}{\sum_{S_j} a^{\sum_k S_{kj}}} \\ &\cdot \prod_j \left(\prod_{k: S_{kj}=1} \frac{\Gamma(\sum_m \alpha_\theta)}{\sum_m \Gamma(\alpha_\theta)} \prod_{m=1}^{M} \theta_{k,j,m}^{\alpha_\theta - 1} \cdot \prod_{k: S_{kj}=0} \frac{\Gamma(\sum_m \alpha_\theta)}{\sum_m \Gamma(\alpha_\theta)} \prod_{m=1}^{M} \theta_{k,j,m}^{\alpha_\theta - 1} \right) \end{aligned} \tag{3.6}$$

3.3.3 Sampling Methods

Denote $\vec{H}(\cdot)$ as a function that returns the count of each category in the supplied vector. Let $(\vec{a})^{\vec{b}}$ stand for the vectorized power function, which raises the elements of $\vec{a}$ to the power of the corresponding elements of $\vec{b}$, respectively.

There are three parameters in our model: $\mathbf{Z}$, $\mathbf{S}$, Θ. We will iteratively sample $\mathbf{Z}$, Θ, and $\mathbf{S}$ as follows:

1. Sample Z given Y, S, Θ

Conditional on the assignment of all other Z_is, the prior for Z_i is:

$$P(Z_i = k \mid Z^{-i}) = \begin{cases} \frac{\alpha_z}{\alpha_z + I - 1} & \text{if k is a new cluster} \\ \frac{C_k^{-i}}{\alpha_z + I - 1} & \text{if k is an existing cluster} \end{cases} \tag{3.7}$$

For the new cluster, since there is no data available, we can draw Θ and S directly from their prior distributions. Because the parameter space for Z_i is $\{0, 1, \cdots, K\}$, we can calculate the conditional posterior probability for Z_i of taking each of those possible values and sample Z_i from a multinomial distribution proportional to this posterior probability.

$$\begin{aligned} P(Z_i \mid Z^{-i}, \Theta, S, Y) &\propto P(Y \mid Z_i, Z^{-i}, \Theta, S) \cdot P(Z_i \mid Z^{-i}) && (3.8) \\ &\propto \prod_{j=1}^{J} \theta_{j, Z_i, y_{ij}} \cdot P(Z_i \mid Z^{-i}) && (3.9) \end{aligned}$$

2. Sample S given Y, Z, Θ

To sample S, we first integrate out Θ. The conditional posterior distribution for S can then be written as

$$\begin{aligned} P(S_j \mid S^{-j}, Z, Y) &\propto P(Y \mid S_j, S^{-j}, Z) \cdot P(S_j \mid S^{-j}) \\ &\propto B(\underset{i:Z_i=k, S_{kj}=0}{H(Y_{ij})} + \vec{\alpha}_\theta) \cdot \prod_{k:S_{kj}=1} B(\underset{i:Z_i=k}{H(Y_{ij})} + \vec{\alpha}_\theta) \qquad (3.10) \\ &\cdot a^{\sum_k S_{kj}} \end{aligned}$$

where $B(\vec{\alpha})$ is the multivariate beta function as in 3.11 and α_θ is the hyperpa-

rameter of the Dirichlet prior we assign to Θ.

$$B(\vec{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)} \tag{3.11}$$

There are $2^K - K$ different configurations of S_j. We can thus calculate the conditional posterior probability for each configuration of S_j and then draw S_j proportionally.

3. Sample Θ given Y, S, Z

   As a result of the above setting, the multinomial parameters for foreground and background are assumed to be from the same Dirichlet distribution. We will further experiment and demonstrate the sensitivity of the algorithm at the end of this Chapter.

   The conditional posterior for $\theta_{k,j}$ can be written as

$$P(\theta_{k,j} \mid \Theta^{-k,j}, S, Z, Y) \propto P(Y \mid \theta_{k,j}, \Theta^{-k,j}, S, Z) \cdot P(\theta_{k,j} \mid \Theta^{-k,j}) \tag{3.12}$$

   Considering column j, if $S_{kj} = 1$, then cluster k is a distinctive cluster for column j, and we will sample $\theta_{j,S_j,k}$ based on the data only belongs to cluster k; alternatively, if $S_{kj} = 0$, then cluster k is one of the few identical clusters, and we will sample $\theta_{j,S_j,k}$ based on the combined data that belongs to those clusters. This is done only once for each of the identical clusters and the same

sampled value of $\theta_{j,S_j,k}$ is assigned to each of them.

$$\underset{S_{kj}=1}{\theta_{k,j}} \sim Dirichlet(\underset{i:Z_i=k}{H(Y_{ij})} + \vec{\alpha}_\theta) \tag{3.13}$$

$$\underset{S_{kj}=0}{\theta_{k,j}} \sim Dirichlet(\underset{i:Z_i=m,\ s.t.\ S_{mj}=0}{H(Y_{ij})} + \vec{\alpha}_\theta) \tag{3.14}$$

Let us illustrate the process again using $K = 3$ as an example.

Given $K = 3$, there are $2^3 - 3 = 5$ different distinction patterns for S_j, which are

$$\begin{pmatrix}0\\0\\0\end{pmatrix}\ \begin{pmatrix}1\\0\\0\end{pmatrix}\ \begin{pmatrix}0\\1\\0\end{pmatrix}\ \begin{pmatrix}0\\0\\1\end{pmatrix}\ \begin{pmatrix}1\\1\\1\end{pmatrix}$$

Suppose the configuration of S_j is

$$\begin{pmatrix}0\\1\\0\end{pmatrix}$$

We can see that S_{1j} and S_{3j} are both 0, to sample their frequency probability vector $\theta_{j,S_j,1}$ and $\theta_{j,S_j,3}$, we pool the observations from column j which satisfies $Z_i = 1$ or 3, denote this data as $\vec{D}_0$. Then we sample

$$\vec{p}_0 \sim Dirichlet\big(\vec{H}(\vec{D}_0) + \vec{\alpha}_\theta\big)$$

We assign the value of $\vec{p}_0$ to $\theta_{j,S_j,1}$ and $\theta_{j,S_j,3}$.

Notice that $S_{2j} = 1$, which means cluster 2 is from a different distribution than cluster 1 and 3. Thus we only use the observations from column j that satisfy $Z_i = 2$, denote this data as $\vec{D}_2$. Then we sample

$$\vec{p}_1 \sim Dirichlet\big(\vec{H}(\vec{D}_2) + \vec{\alpha}_1\big)$$

We assign the value of $\vec{p}_1$ to $\theta_{j,S_j,2}$.

Further, if the configuration of S_j is

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

We can see that S_{1j}, S_{2j} and S_{3j} are all 1, which means cluster 1, cluster 2 and cluster 3 are from different distributions respectively. To sample $\vec{\theta}_{j,S_j,1}$, we only use the observations from column j that satisfy $Z_i = 1$, denote this data as $\vec{D}_1$, and sample

$$\vec{p}_1 \sim Dirichlet\big(\vec{H}(\vec{D}_1) + \vec{\alpha}_\theta\big)$$

We assign the value of $\vec{p}_1$ to $\theta_{j,S_j,1}$;

Next we use the observations from column j that satisfy $Z_i = 2$, denote this data as $\vec{D}_2$, and sample

$$\vec{p}_2 \sim Dirichlet\big(\vec{H}(\vec{D}_2) + \vec{\alpha}_\theta\big)$$

We assign the value of $\vec{p}_2$ to $P_{j,S_j,2}$; similarly, we can obtain the sample for $P_{j,S_j,3}$.

We do this according to the distinction pattern of S_j and for every column j and obtain a Θ matrix of K by J by M.

3.3.4 Determination of Number of Clusters

The number of clusters K is incorporated into our model. However, sometimes the algorithm will be trapped into local mode because of the high energy barrier between settings of different clusters. From our simulation study, it is difficult to increase the number of clusters, but the number of clusters will converge to the true number of clusters if starting with a higher number. In our model, the joint posterior probabilities for different numbers of clusters are up to a normalizing constant and are thus comparable.

We will run independent chains starting with different number of clusters until the number of clusters converges to a stable number. Then we will compare the joint posterior mode for different numbers of clusters in determining the optimal number of clusters.

3.3.5 Algorithm Summary

1. Start with $K = 2$.

2. Arbitrarily set values of $Z^{(1)}$ and $S^{(1)}$ in the Gibbs algorithm.

3. Suppose we have already obtained $Z^{(t)}$ and $S^{(t)}$, update Θ, Z, S by taking the

following steps:

(a) Sample $\Theta^{(t+1)}$ from Y, $S^{(t)}$ and $Z^{(t)}$.

(b) Sample $Z^{(t+1)}$ from Y, $S^{(t)}$ and $\Theta^{(t+1)}$.

(c) Sample $S^{(t+1)}$ from Y, $Z^{(t+1)}$ and $\Theta^{(t+1)}$.

(d) $t \to t+1$.

4. Iteratively run Step 3 until convergence.

5. Record the output at the joint posterior mode.

6. $K \to K+1$ until the number of clusters converges.

7. Select the value of K that leads to the highest joint posterior probability as the optimal number of clusters. Output the posterior samples of Z, Θ and S under this K.

3.4 Simulation Study

3.4.1 Validation of Biclustering Results

We used a Jaccard Index (Jaccard (1901)) to validate the correctness of our biclustering results. The Jaccard Index is an effective measure for comparing the similarity and diversity between sample sets. For any two sets S_1 and S_2, their Jaccard Index is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(S_1, S_2) = \frac{\mid S_1 \cap S_2 \mid}{\mid S_1 \cup S_2 \mid}$$

In our case, where there is more than one bicluster in each set, we use an adapted version of Jaccard Index (Kaiser and Leisch (2008)) to measure the similarity of our estimated bicluster structure and the true structure.

$$J(E,T) = \frac{1}{K_1}\sum_{a=1}^{K_1}\sum_{b=1}^{K_2}\frac{\mid E_a \cap T_b \mid}{\mid E_a \cup T_b \mid}$$

where K_1 is the number of biclusters in the estimated set, K_2 is the number of biclusters in the true set; E_a is the a_{th} bicluster in the estimated set; T_b is the b_{th} bicluster in the true set.

The value of a Jaccard Index of any two sets is between 0 and 1. Larger values suggest higher similarity between two bicluster sets.

3.4.2 Date Generation

For the simulation study, we generated three data matrices. Data set A is a 600 by 2000 matrix, which contains 3 biclusters. Data set B is a 1000 by 1400 matrix, which contains 5 biclusters. Data set C is a 600 by 600 matrix containing 3 biclusters. The number of categories in Data set A was set to be 3, i.e. $M = 3$. For background columns, the samples were drawn from a multinomial distribution with probability vector $\{0.33, 0.33, 0.33\}$. For foreground, the samples were drawn from multinomial with probability vector $\{0.1, 0.3, 0.6\}$, $\{0.5, 0.1, 0.4\}$ and $\{0.7, 0.1, 0.2\}$ respectively; The number of categories in Data set B was set to be 4, i.e. $M = 4$. For columns, the multinomial probability vector for both background and foreground samples were independent draws from Dirichlet distribution with parameter $\{1, 1, 1, 1\}$. Data set C was generated in the way that all columns were random draws from multinomial

distribution $\{0.33, 0.33, 0.33\}$. According to our bicluster definition, it has 1 bicluster and S is the same over all columns. The number of categories of the data in Data set C was set to 3.

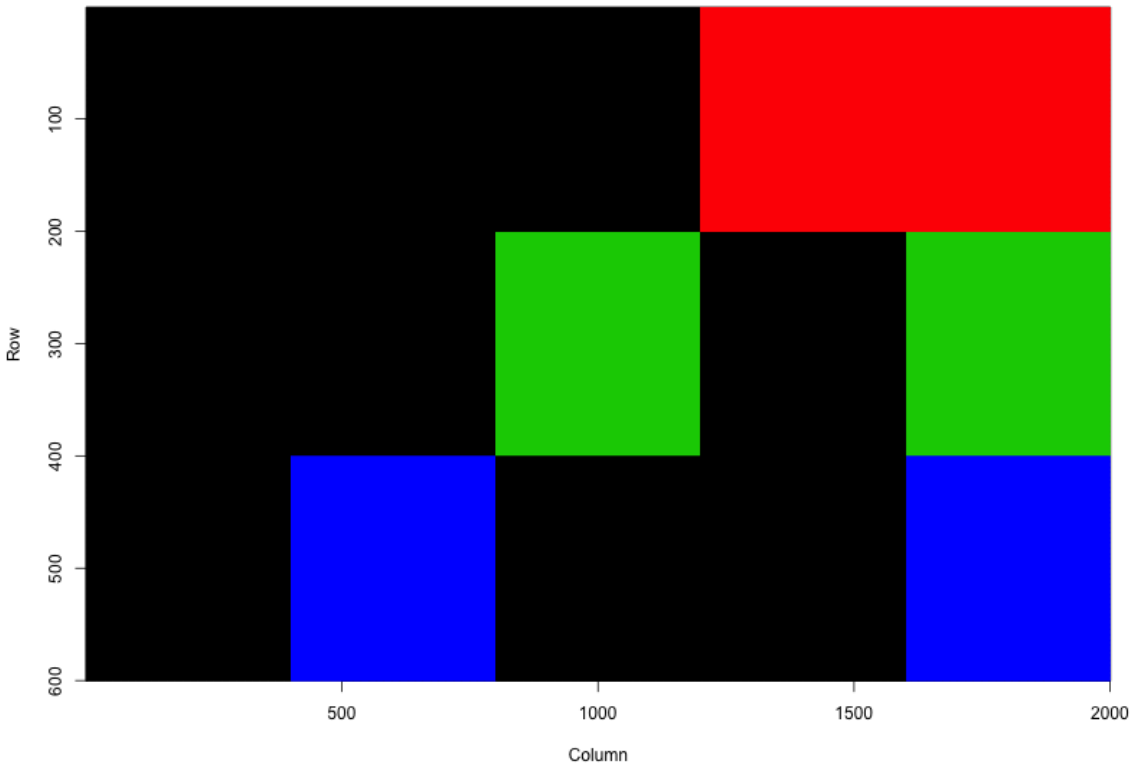

Figure 3.1: The true bicluster structure from which Data matrix A is simulated. Each color demonstrates a different bicluster. Each of the 3 biclusters contains 200 rows and 800 columns.

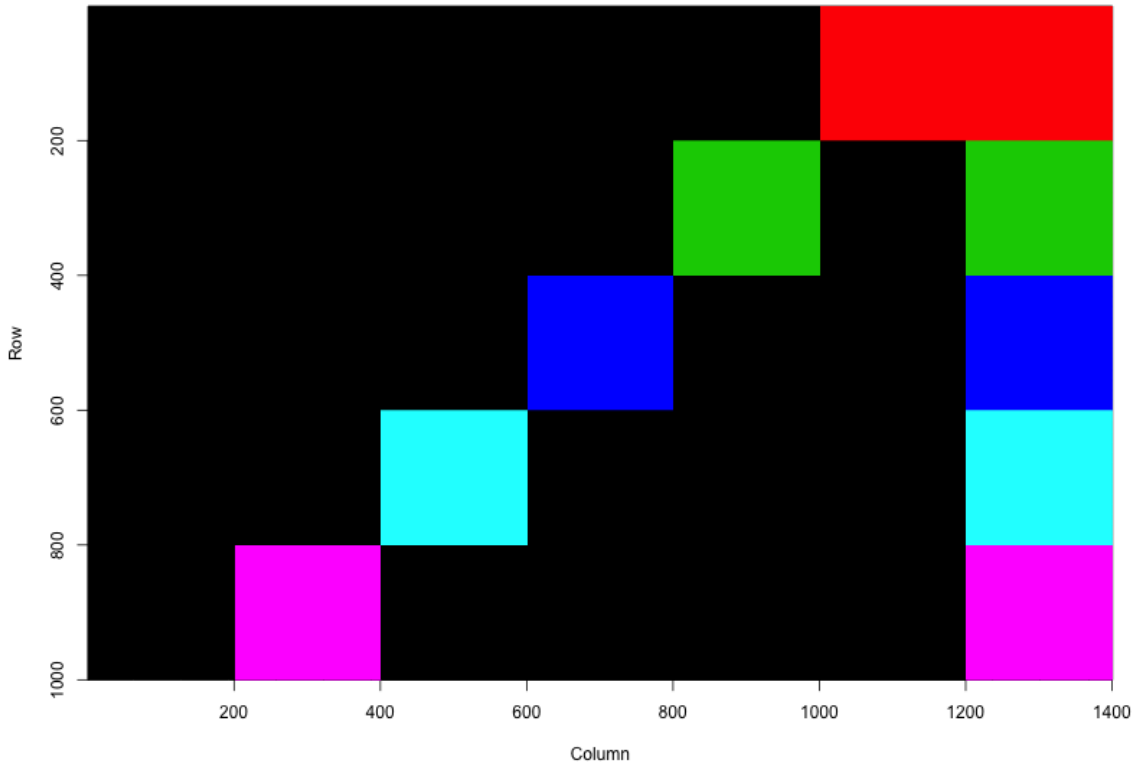

Figure 3.2: The true structure for Data matrix B. There are 5 biclusters embedded, with an equal size 200 by 400.

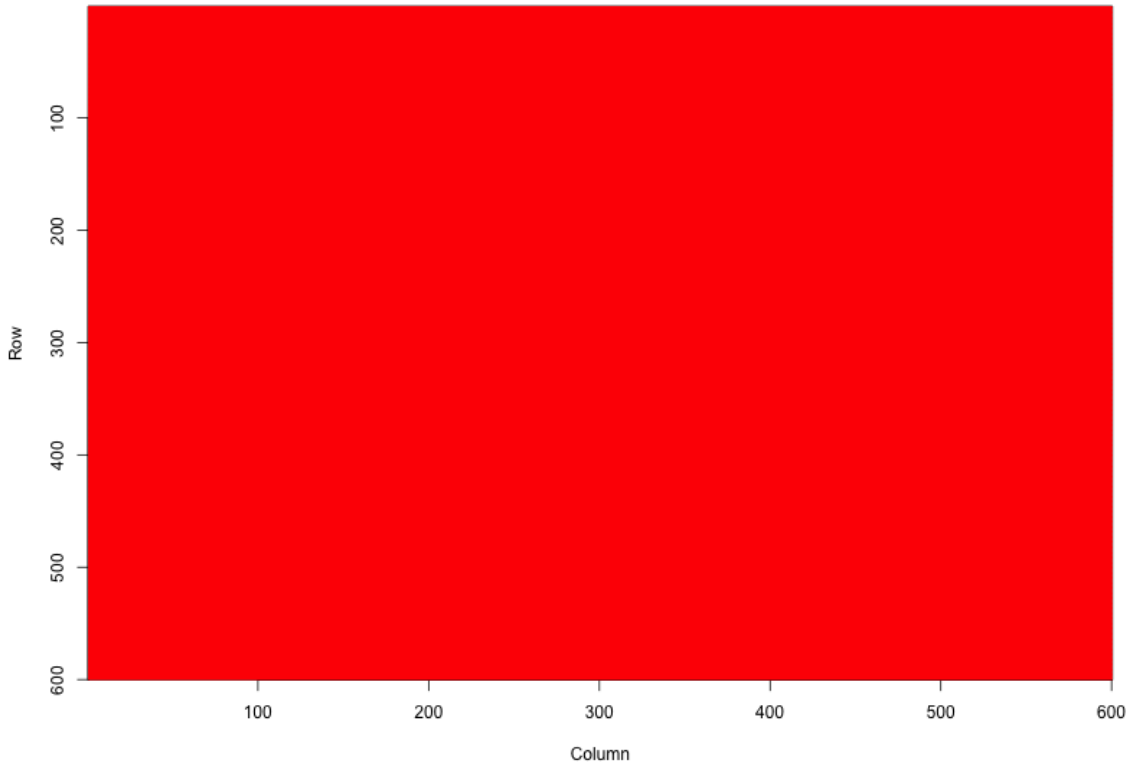

Figure 3.3: The true structure for Data matrix C. The data are generated randomly for each category. It corresponds to 1 background bicluster in our definition

Notice that in the first data set, there are a total of $2^3 - 3 = 5$ distinctive patterns that S can take and we embed all those patterns into the data, as seen in Figure 3.1. In the second data set, there are a total of $2^5 - 5 = 27$ distinctive patterns S can take and we embeded 7 of them, which is displayed in Figure 3.2. In the thrid data set, S is 0 across all columns as in Figure 3.3.

3.4.3 Data set A: 3 clusters

Following the sampling procedure presented in section 3.3.5, we started with 3 clusters and let the algorithm run for 1,000 steps and discard the initial 500 steps as burn-in. The priors were set as $\alpha_z = \{1, 1, 1\}$, $\alpha_\theta = \{2, 2, 2, 2\}$, $a = 0.05$. The initial values for Z, S, and Θ were assigned randomly based on their priors. The trace plot of joint posteriors for all 3 chains can be viewed in Figure 3.4. We can see that chain 2 was trapped in local mode. To further investigate, we increased the number of independent chains to 100 and found out that 11 of the chains were stuck in local modes.

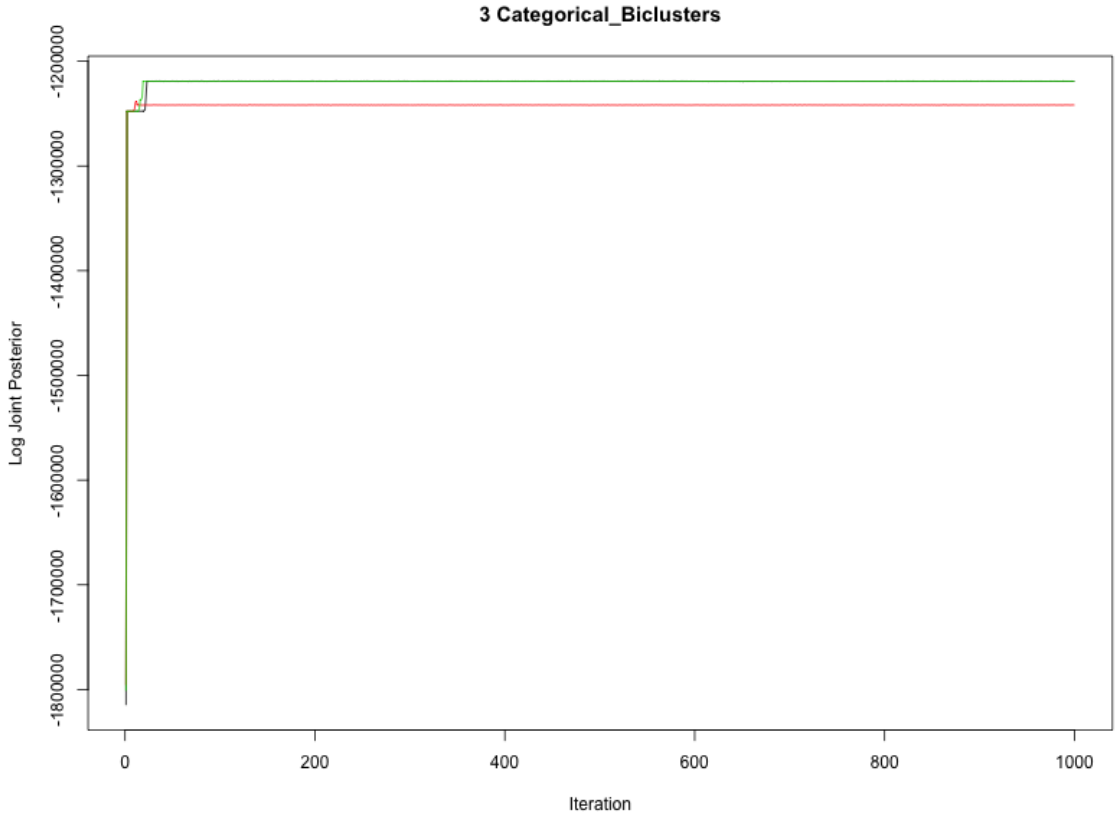

Figure 3.4: Trace plot for the joint posteriors of 3 independent chains. Chain 2 is plotted in red.

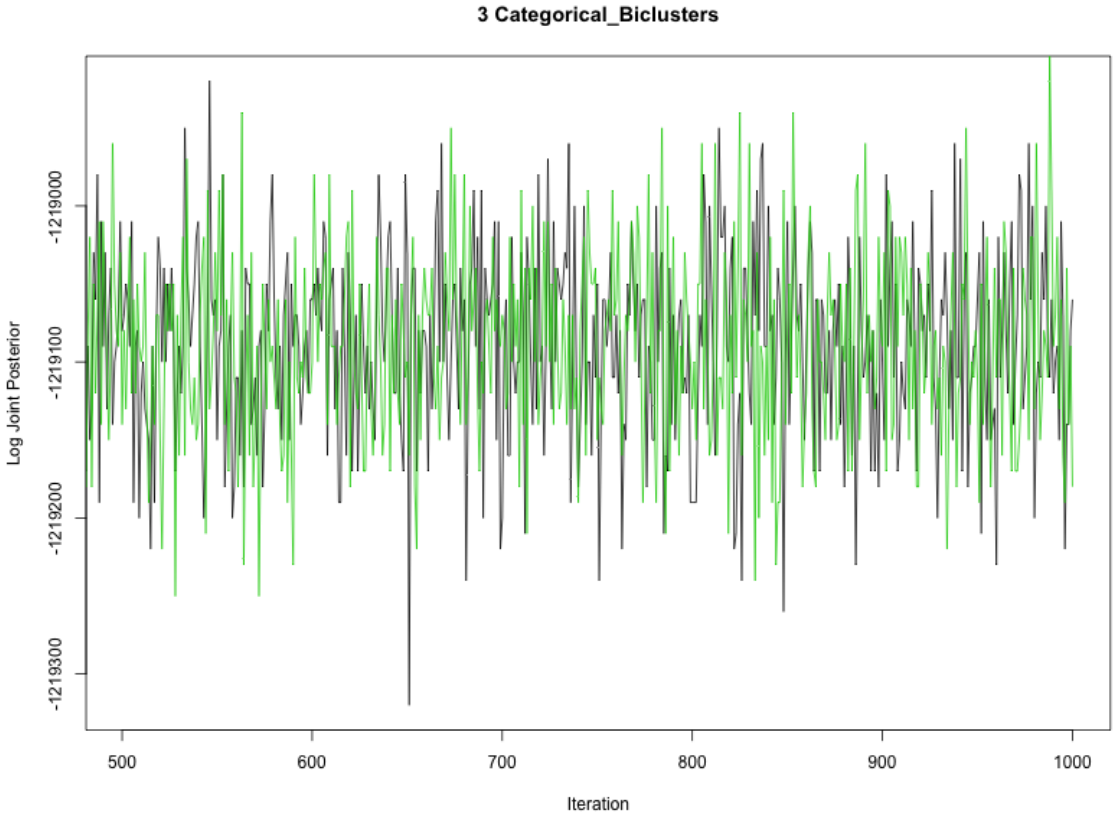

Figure 3.5: Trace plot for the joint posteriors of the chain 1 and 2, after burn-in.

By looking into chain 1 which is not in a local mode, we can plot the recovered bicluster structure as the iteration goes in Figure 3.6. The bicluster structure in the plots correspond to those at step 20, 22, 25, 27, 50, 80. We can see the algorithm converges to the truth very fast even starting from random initial values.

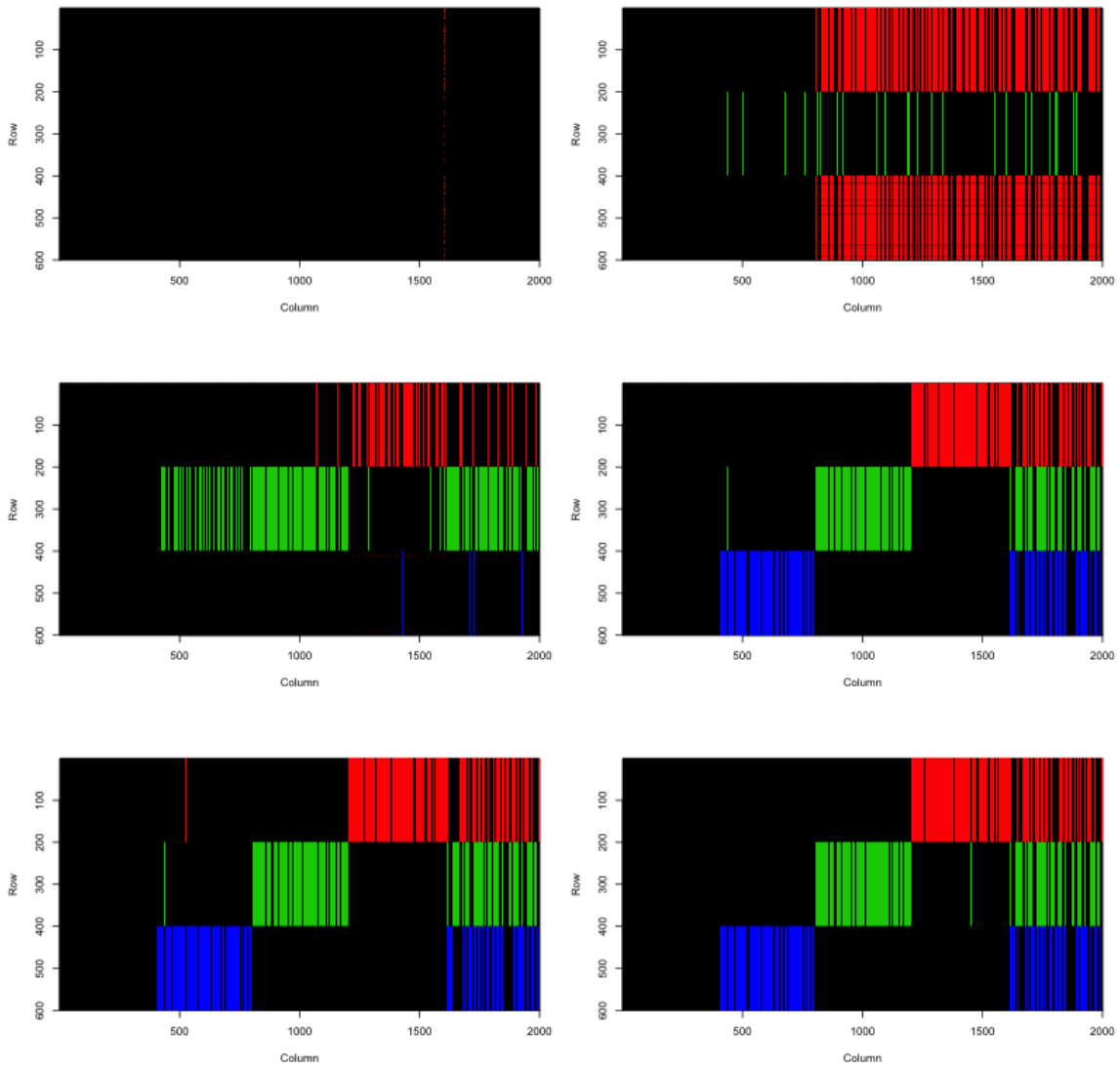

Figure 3.6: Recovered bicluster structure at different step.

As a solution to the local mode trapping problem, we chose the chains with highest joint posteriors and plotted their mixing after burn-in in Figure 3.5. As we can see, the mixing is good. Then we started with 2, 4, 5, 6 clusters, each with 10 independent chains and found out that all these independent runs will eventually converge to 3 or

2 clusters. We compared their joint posterior modes from corresponding chains with highest posteriors and found that the optimal number of clusters in this simulated was 3.

An alternative approach to overcome the local mode problem is to use a hierarchical clustering method to provide a starting configuration for our algorithm. More precisely, we treated the categorical data as continuous and employed the Euclidean distance measure for hiearchical clustering.

We again ran 3 parallel MCMC chains with $\alpha_z = \{1, 1, 1\}$, $\alpha_\theta = \{2, 2, 2, 2\}$, $a = 0.05$. The initial values for Z were set using the results from hierarchical clustering and it converges very fast. The diagnostics for the parallel chains are presented in Figure 3.7 and 3.8. The ACF plot in Figure 3.9 reveals that the autocorrelation drops to 0 at a high rate. The implementation of hierarchical clustering was adapted from open source clustering package Cluster 3.0. (de Hoon et al. (2004))

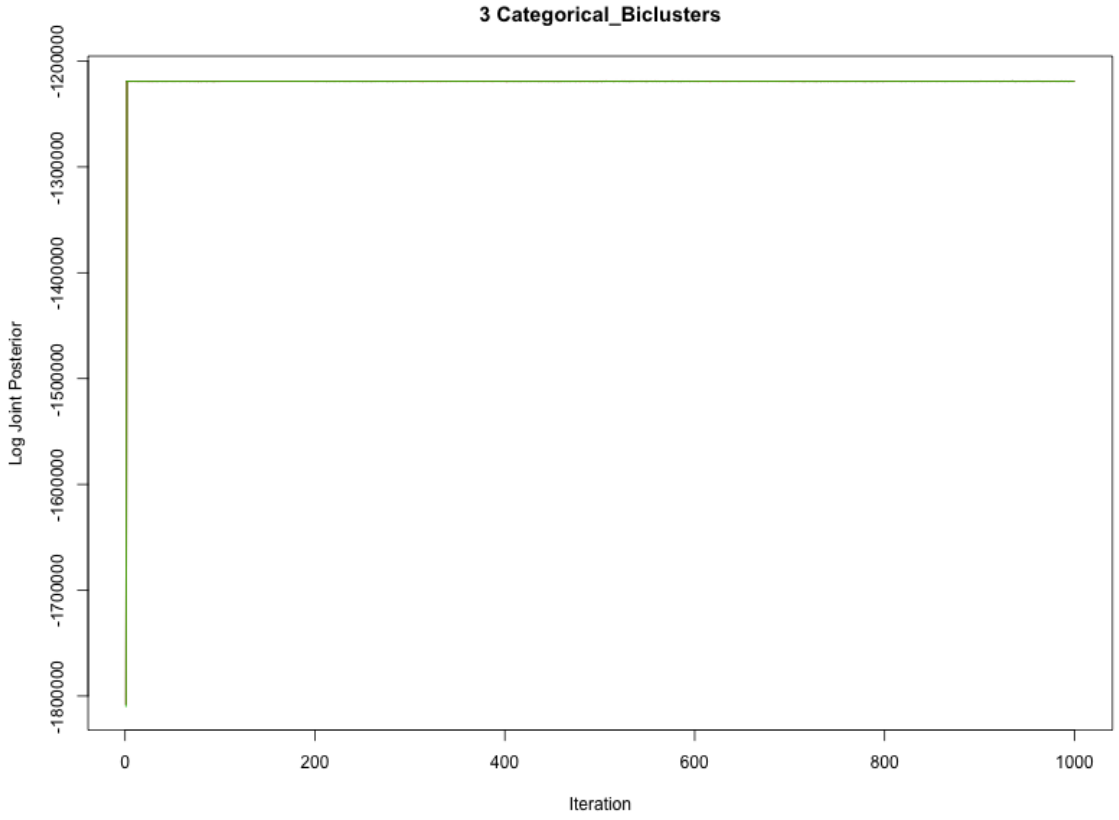

Figure 3.7: Trace plot for the joint posteriors of 3 independent chains, started with hierarchical clustering results.

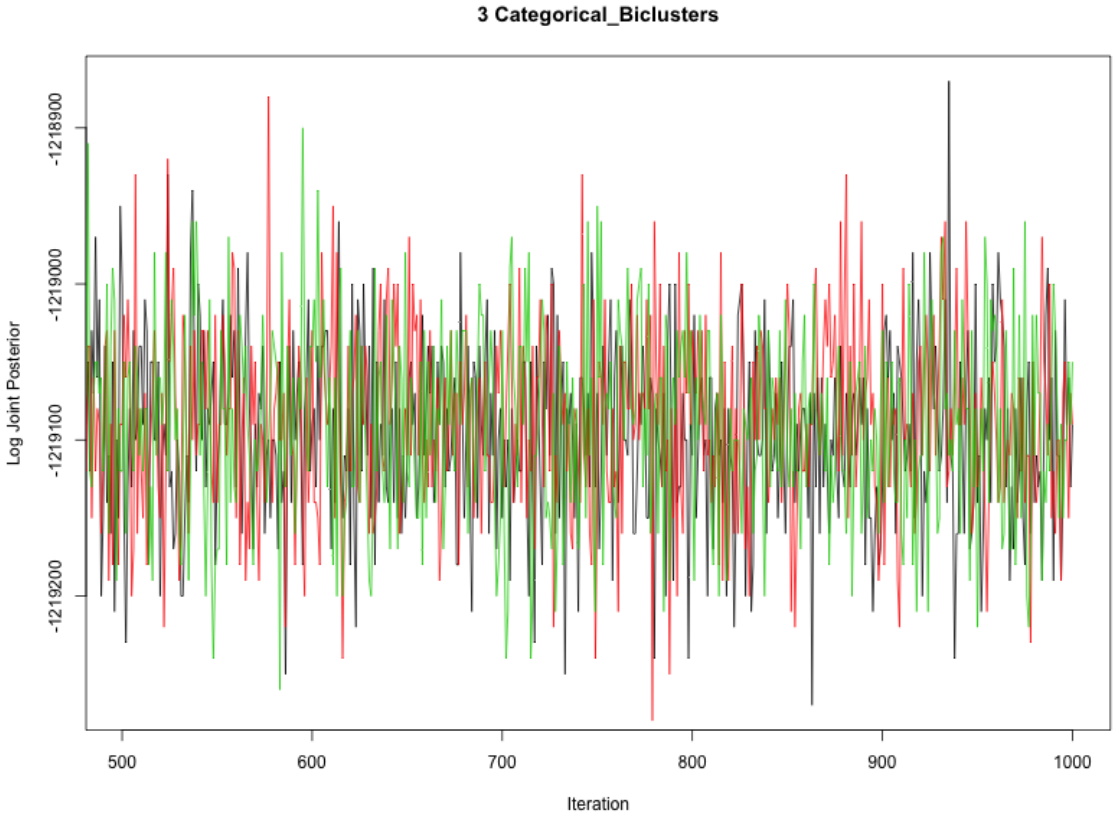

Figure 3.8: Trace plot for the joint posteriors of 3 independent chains after burn-in, started with hierarchical clustering results.

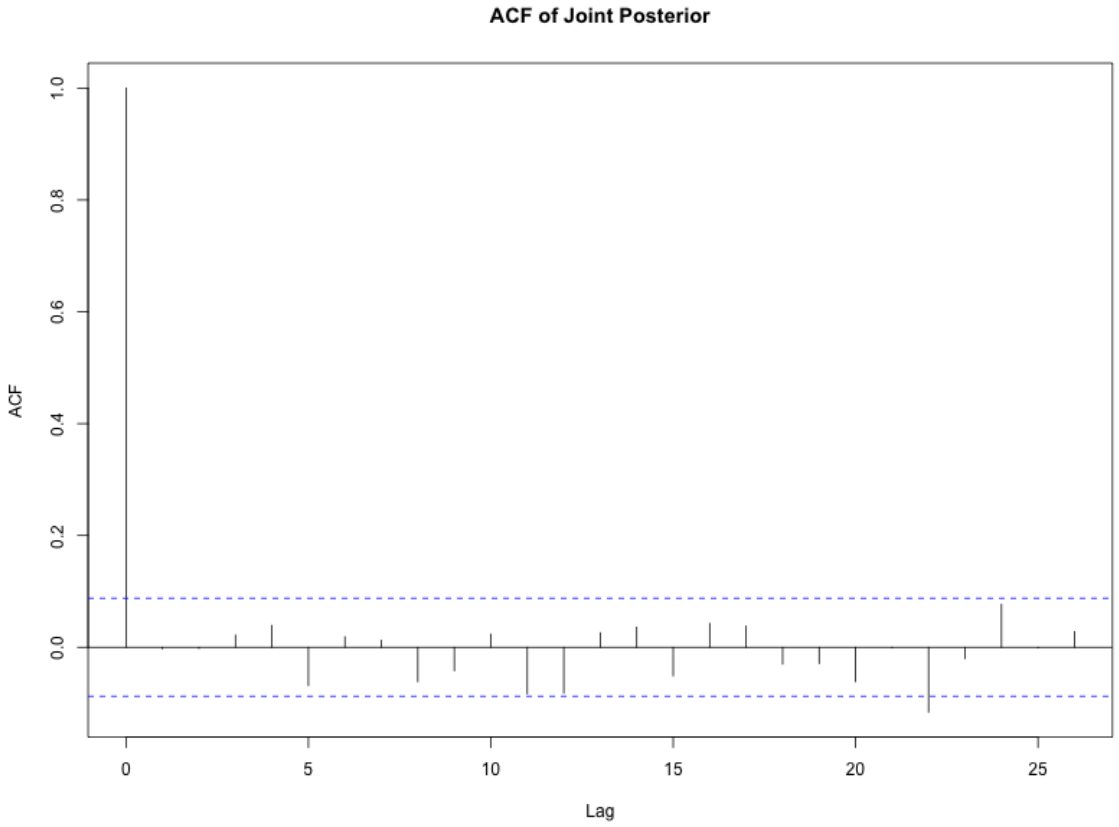

Figure 3.9: ACF plot for the joint posteriors, started with hierarchical clustering results

By running the algorithm with 3, 4, 5, and 6 clusters, the algorithm will all converge to 3 clusters as the iterations run. However, if we started with 2 clusters, the algorithm will stay at 2 because of the high energy barrier. After comparing the joint posterior mode at 2 and 3 clusters, we finally identified 3 as the optimal number of clusters, which is consistent with the true number of clusters. The change of number of clusters from different runs is illustrated in Figure 3.10.

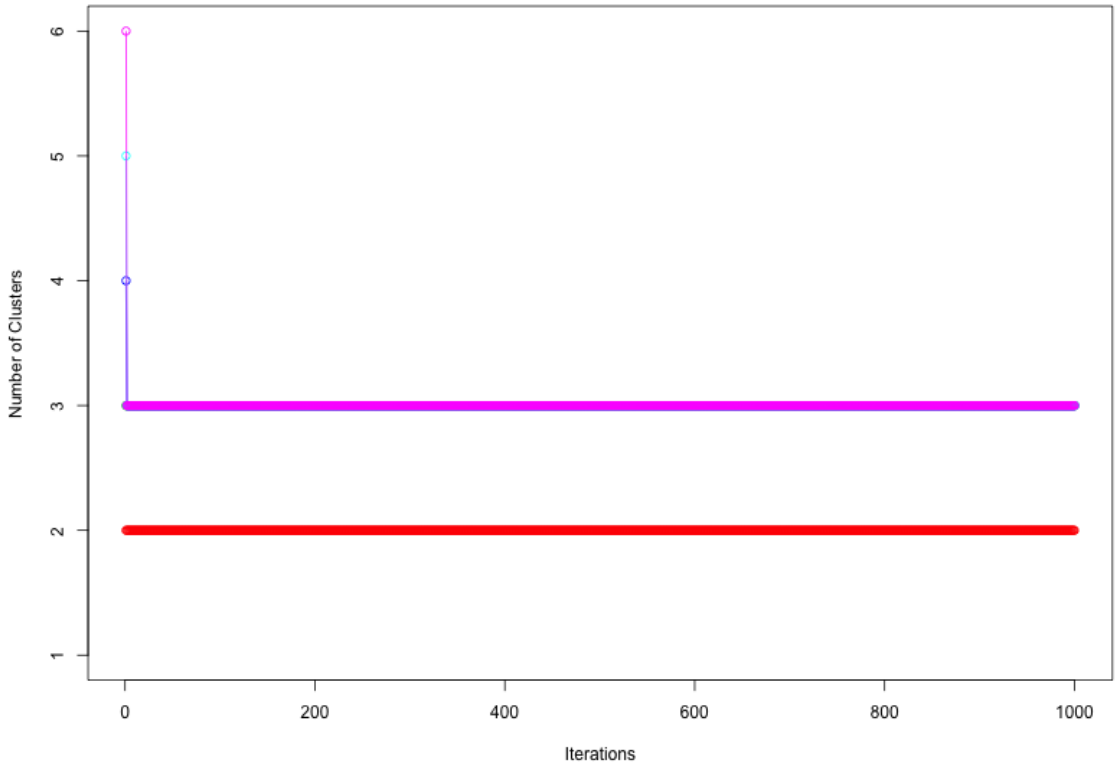

Figure 3.10: Change of number of clusters from independent runs starting from 2, 3, 4, 5, 6 clusters

The results of the recovered biclustering structure is plotted in Figure 3.11. The Jaccard Index for this estimate is 0.87, which indicates a good estimate of the original biclustering structure. Different reasonable settings of priors have also be tested and they have little impact on the estimation accuracy. A detailed comparision of the sensitivity of the algorithm is presented at the end of this Chapter.

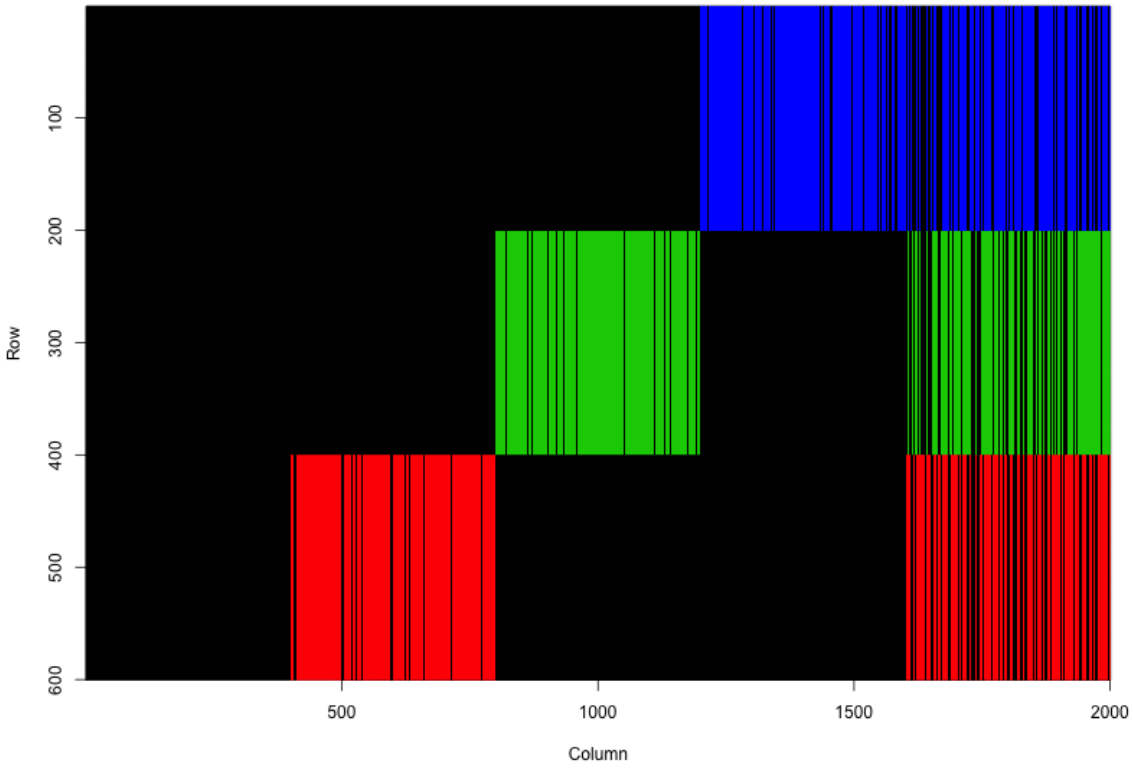

Figure 3.11: the biclusters learned through the MCMC algorithm. They resemble the true structure in general except for a small amount of errors.

3.4.4 Data set B: 5 clusters

We performed similar analysis for Data matrix 2, started with 2, 3, 4, 5, 6, 7, and 8 clusters independently, and the number of clusters converged to 5 when greater than 5. By comparing their joint posterior mode we chose 5 as the optimal number of clusters. The diagnostic plots are presented as in Figure 3.12 and Figure 3.13.

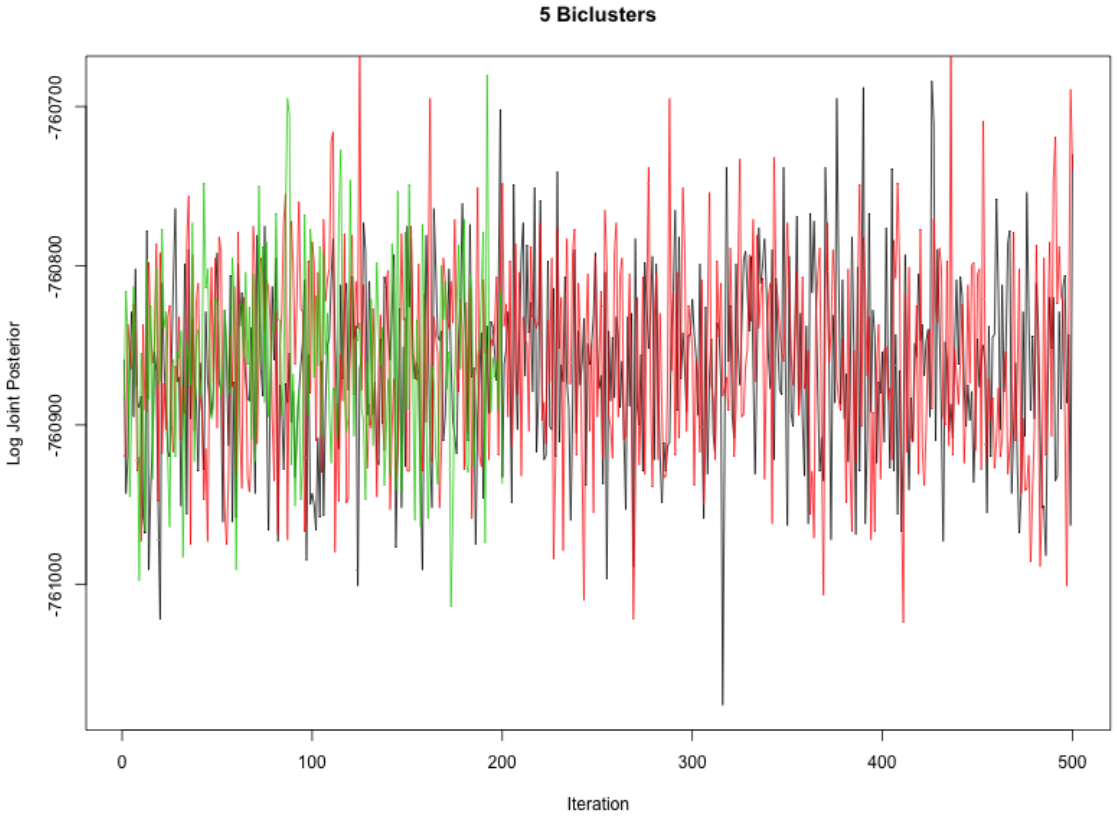

Figure 3.12: Trace plot for the joint posterior of three parallel chains for 5 clusters.

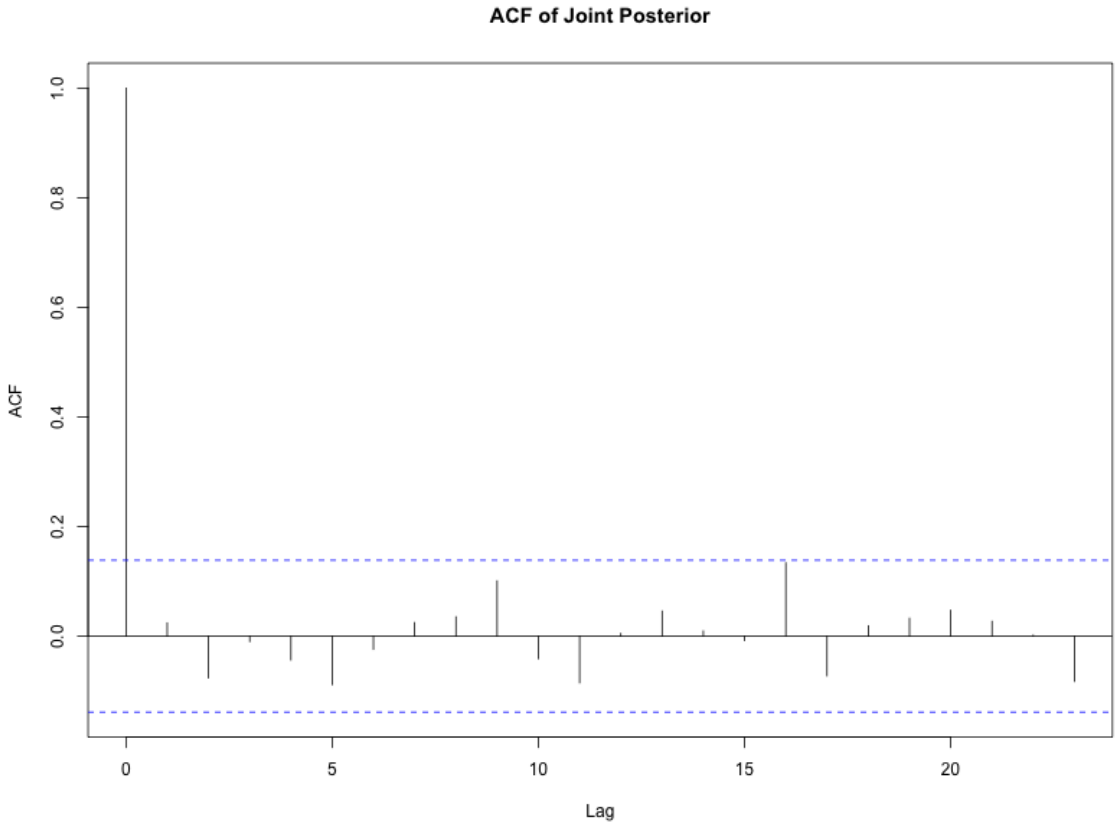

Figure 3.13: Autocorrelation plot for the joint posterior for 5 clusters. No significant autocorrelation emerges when the lag is greater than 1.

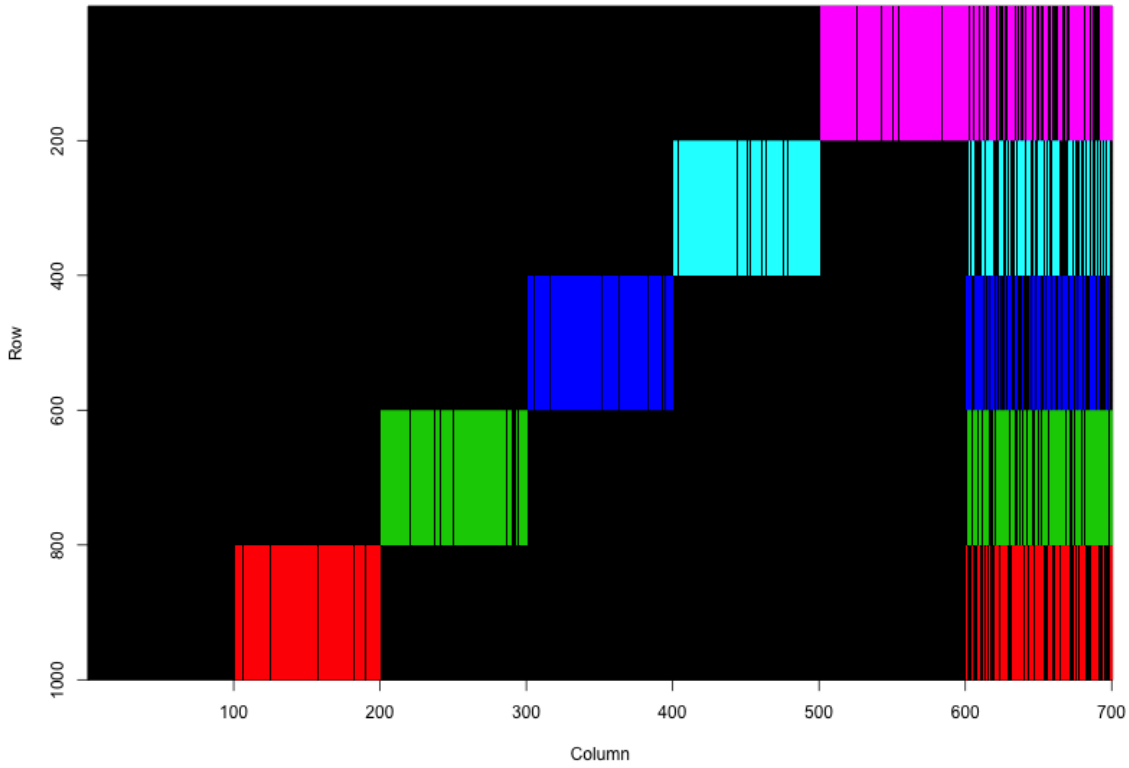

Figure 3.14: The 5 biclusters determined by joint posterior mode. The estimated bicluster structure is very close to the true structure.

The estimated biclustering structure is presented in Figure 3.14. The recovered structure is very similar to the embeded true bicluster structure with a Jaccard Index of 0.86.

3.4.5 Data set C: 1 cluster

For Data set C, we let the algorithm run starting from 1, 2, 3, 4, 5, 6 clusters and it all converges to 1 bicluster, which is the same as we simulated.

3.4.6 Sensitivity tests for Bayesian Categorical BiClustering Model

In this section, we will present the experiments we performed to test the sensitivity of the Bayesian Categorical Biclustering algorithm. To test the sensitivity of the algorithm, we embeded 3 biclusters into the data as in Figure 3.1, each with different similarities between the column bicluster specific multinomial distributions. The number of categories of data was set to be 3. In the simulated data, the same color in each column means the data was drawing from the same multinomial distribution. Different colors on the same column means that they are each from a different multinomial distribution. All the multinomial distributions were generated from the same prior Dirichlet distribution.

As α increases, the sampled probability vectors for column bicluster-specific multinomial distributions, from $Dirichlet(\alpha, \alpha, \ldots, \alpha)$ will tend to be more and more similar to each other. We let α go from 0.1 to 10 and calculated the estimate accuracy by evaluating the Jaccard Index for each simulation. In our Gibbs sampling, we used the same prior setting ($\alpha_z = 1$, $\alpha_p = 2$ and $a = 0.1$) across all tests. The recovered bicluster structures are listed below and the results are presented in Table 3.1.

Table 3.1: Sensitivity Test Results

Dirichlet Prior: α	**a**	α_z	α_p	Jaccard Index
(0.1, 0.1, 0.1)	0.1	1	2	0.779
(0.2, 0.2, 0.2)	0.1	1	2	0.833
(0.3, 0.3, 0.3)	0.1	1	2	0.849
(0.4, 0.4, 0.4)	0.1	1	2	0.846
(0.5, 0.5, 0.5)	0.1	1	2	0.843
(1, 1, 1)	0.1	1	2	0.802
(5, 5, 5)	0.1	1	2	0.601
(7, 7, 7)	0.1	1	2	0.330
(10, 10, 10)	0.1	1	2	0.137

As we can see, when the *Dirichlet* prior for generating the data goes beyond 5, the estimate accuracy drops fast and becomes unsatisfactory. The algorithm is in general very robust. We tried different values for α_z and α_p in our sampling and the recovered bicluster structure and Jaccard Index accuracy are similar. There are lots of potential applications of this algorithm in real life and we hope it can help facilitate new scientific discoveries.

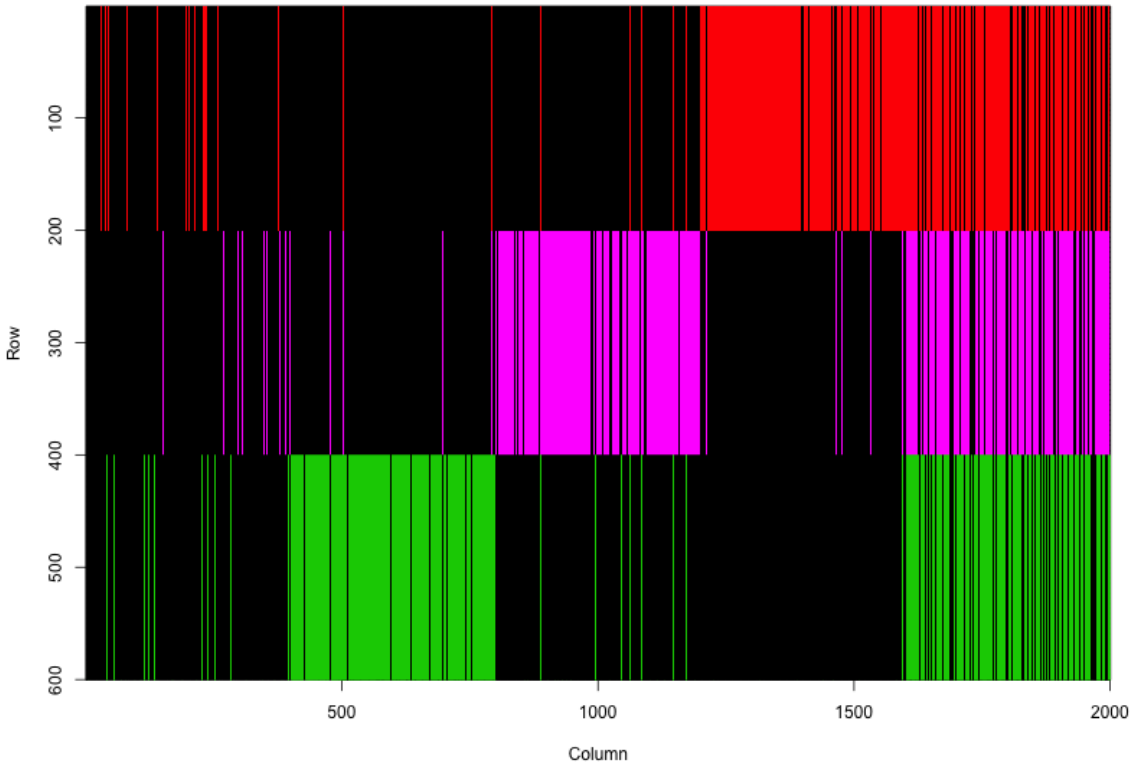

Figure 3.15: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(0.1, 0.1, 0.1)$. Jaccard Index: 0.779

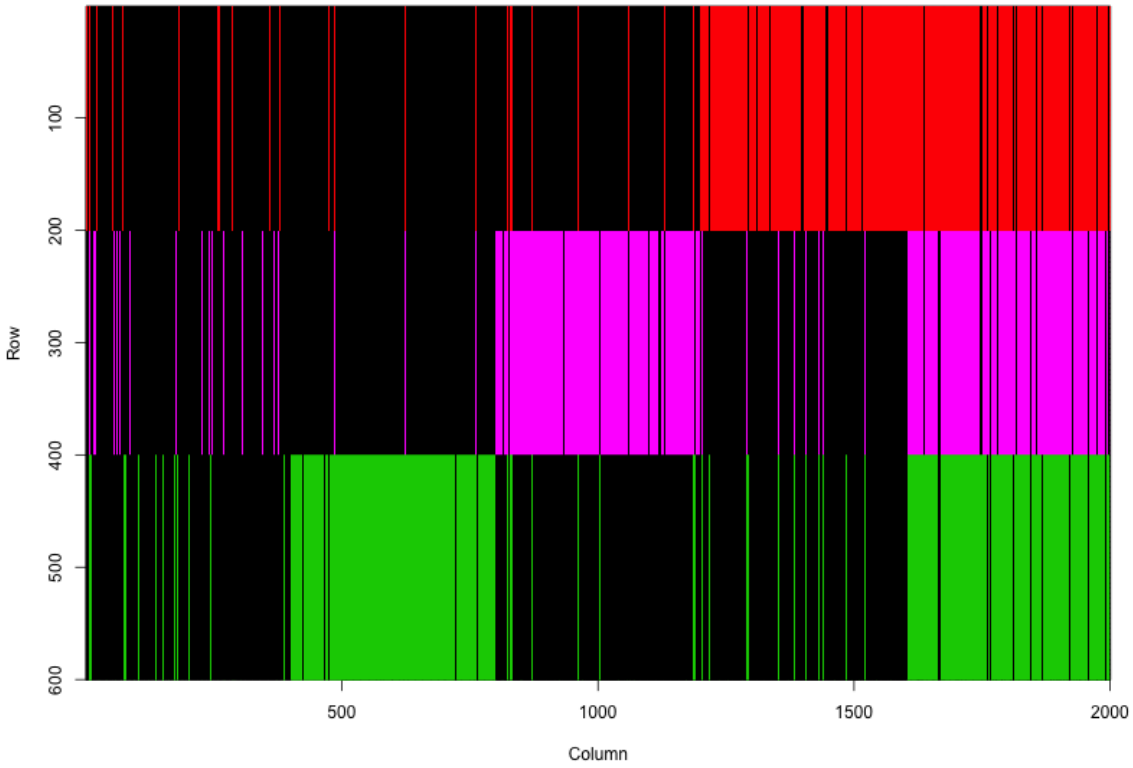

Figure 3.16: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(0.2, 0.2, 0.2)$. Jaccard Index: 0.833

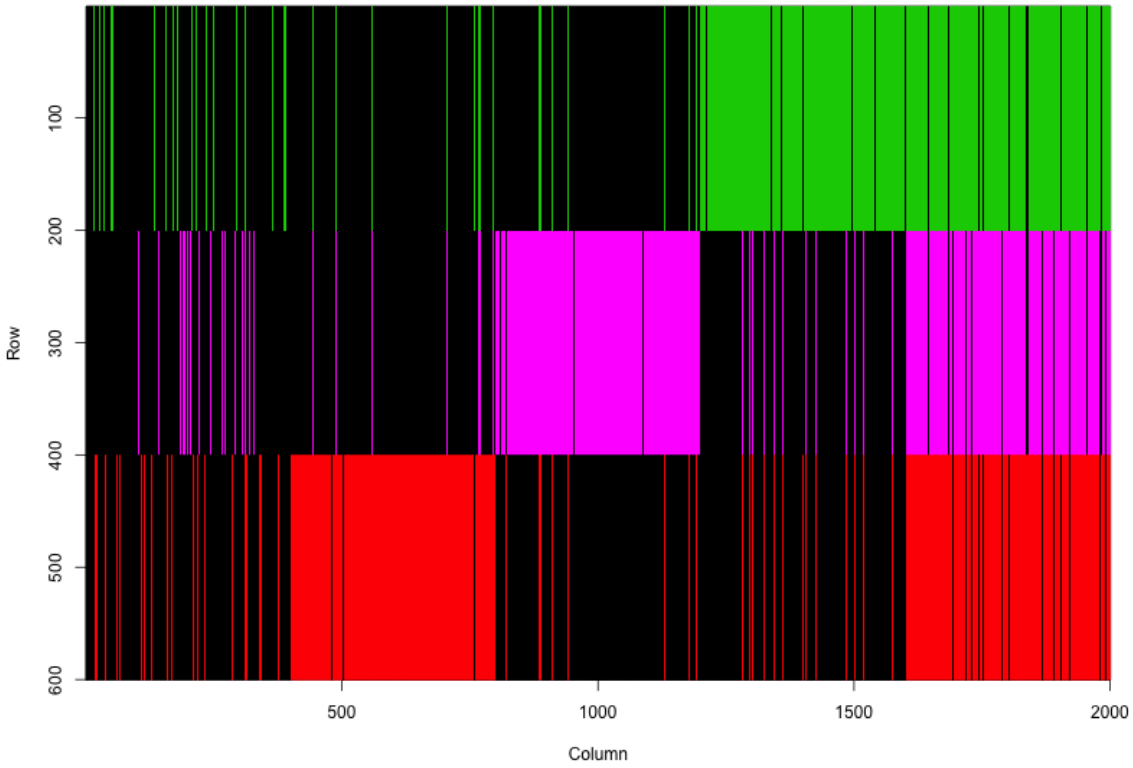

Figure 3.17: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(0.3, 0.3, 0.3)$. Jaccard Index: 0.849

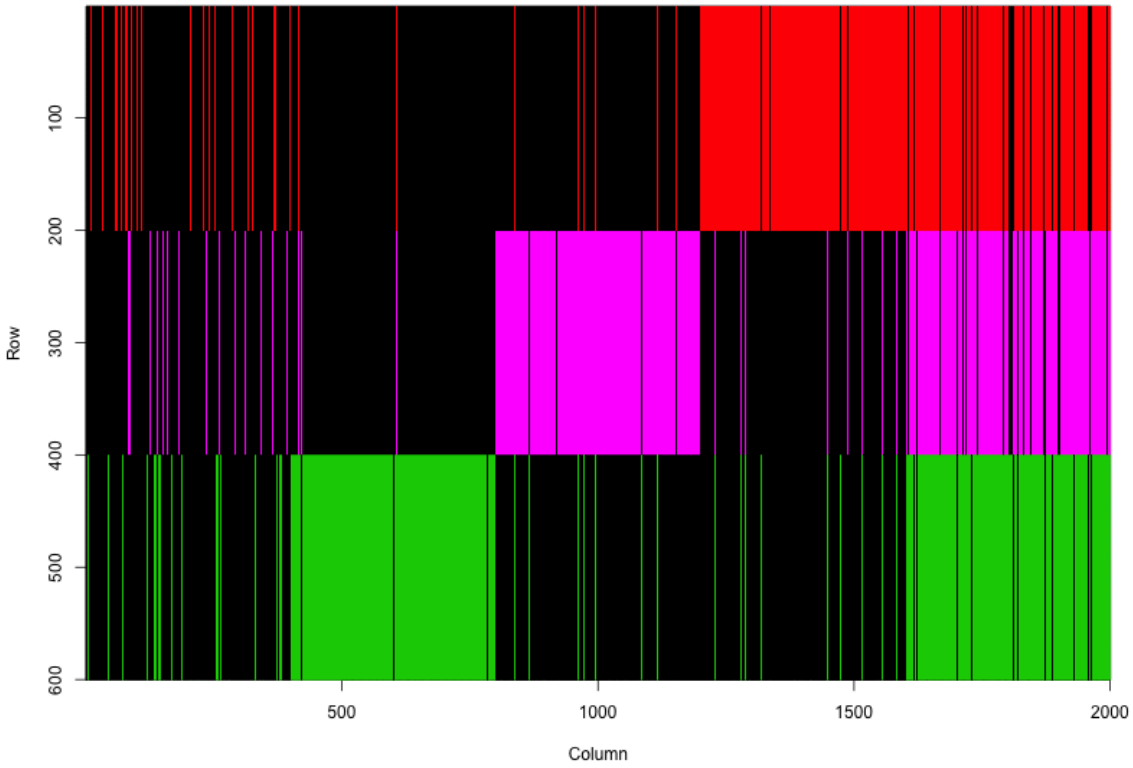

Figure 3.18: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(0.4, 0.4, 0.4)$. Jaccard Index: 0.846

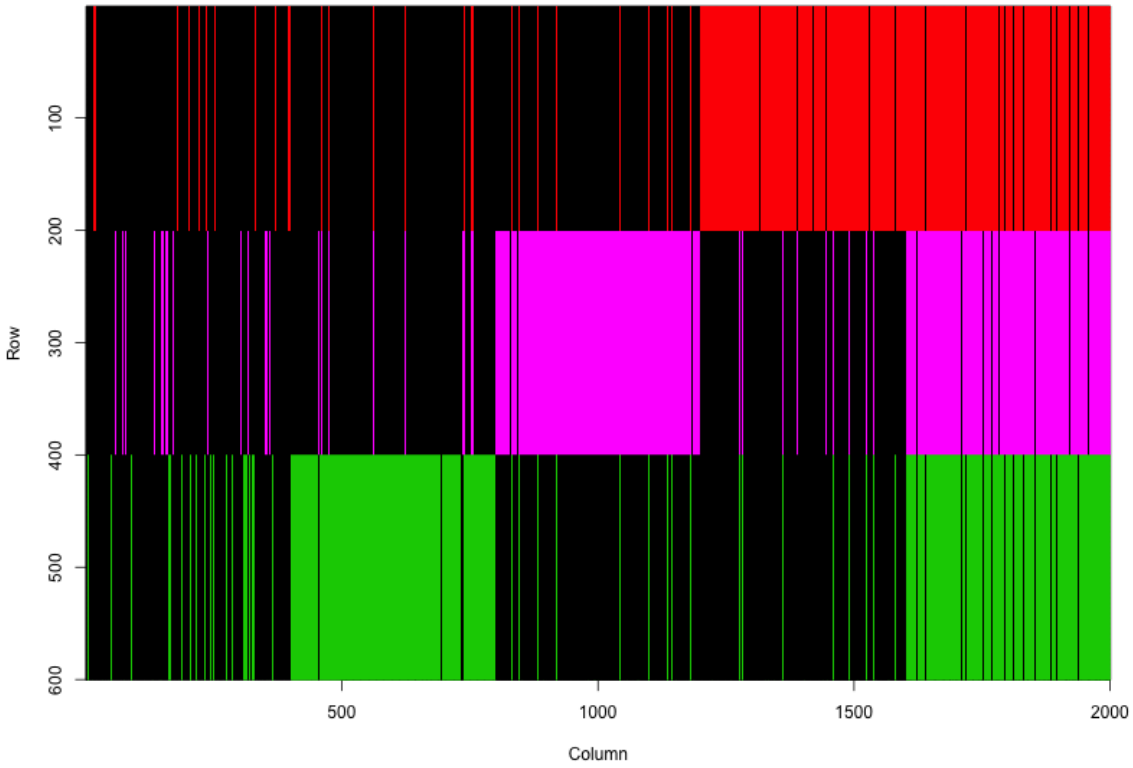

Figure 3.19: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(0.5, 0.5, 0.5)$. Jaccard Index: 0.843

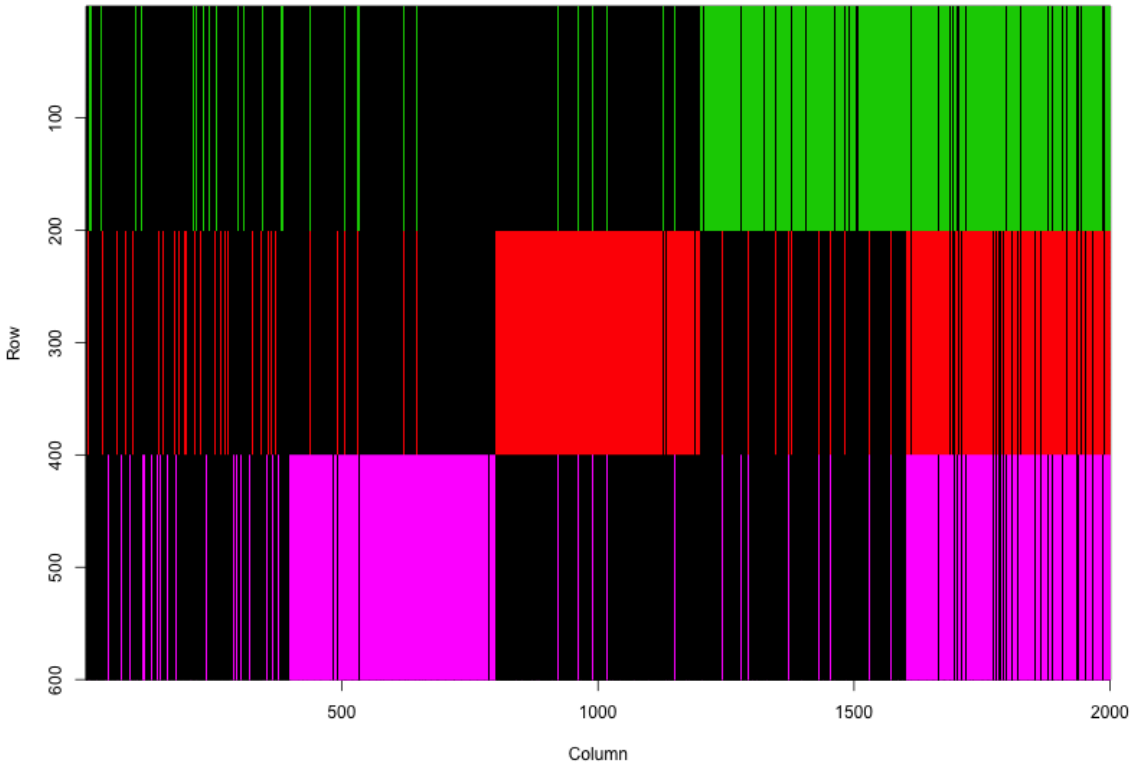

Figure 3.20: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(1, 1, 1)$. Jaccard Index: 0.802

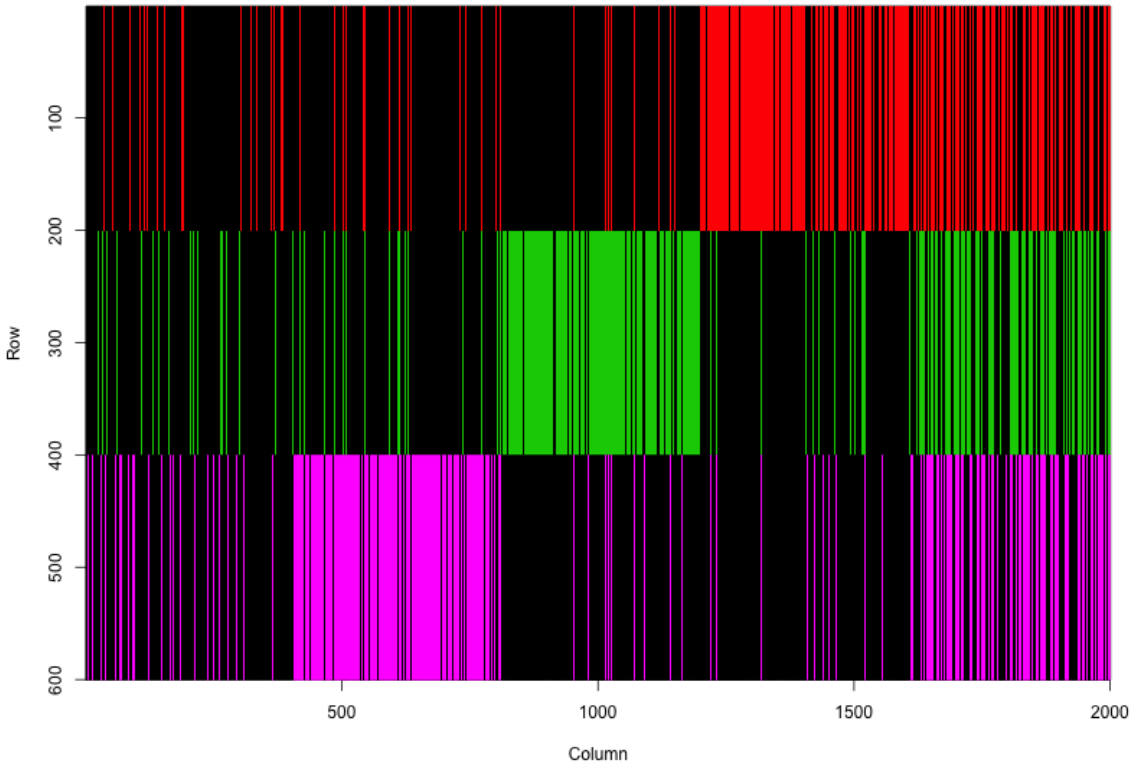

Figure 3.21: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(5, 5, 5)$. Jaccard Index: 0.601

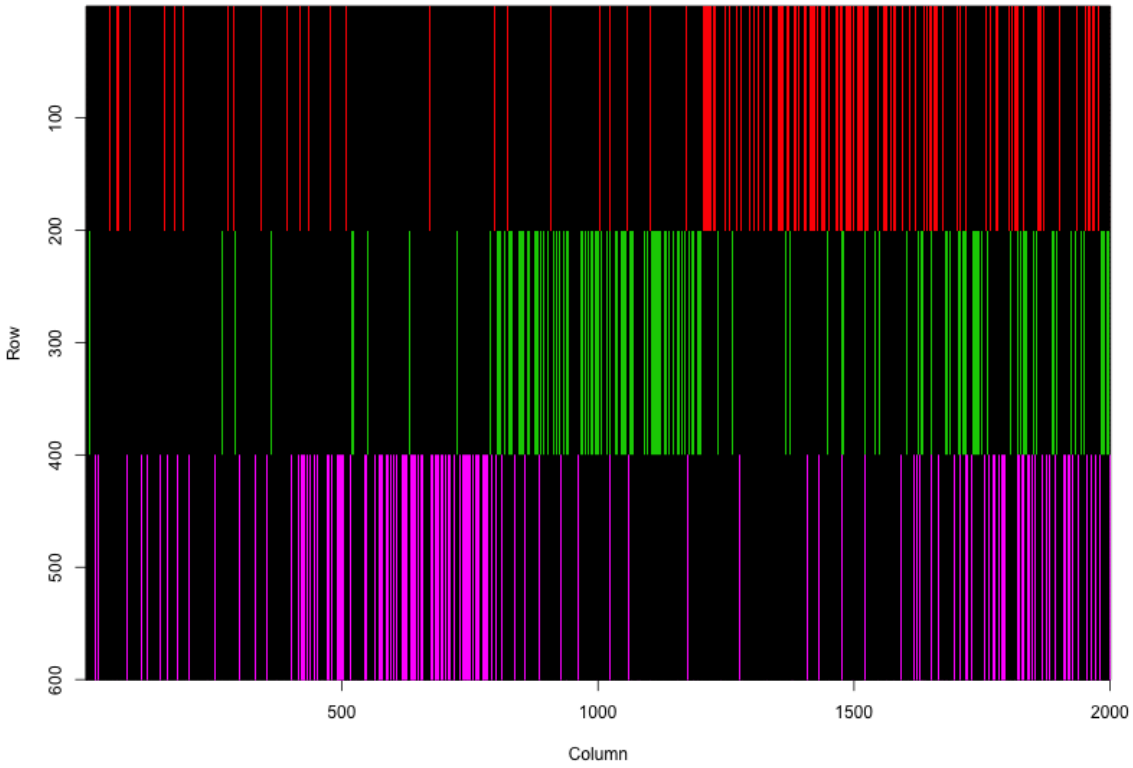

Figure 3.22: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(7, 7, 7)$. Jaccard Index: 0.330

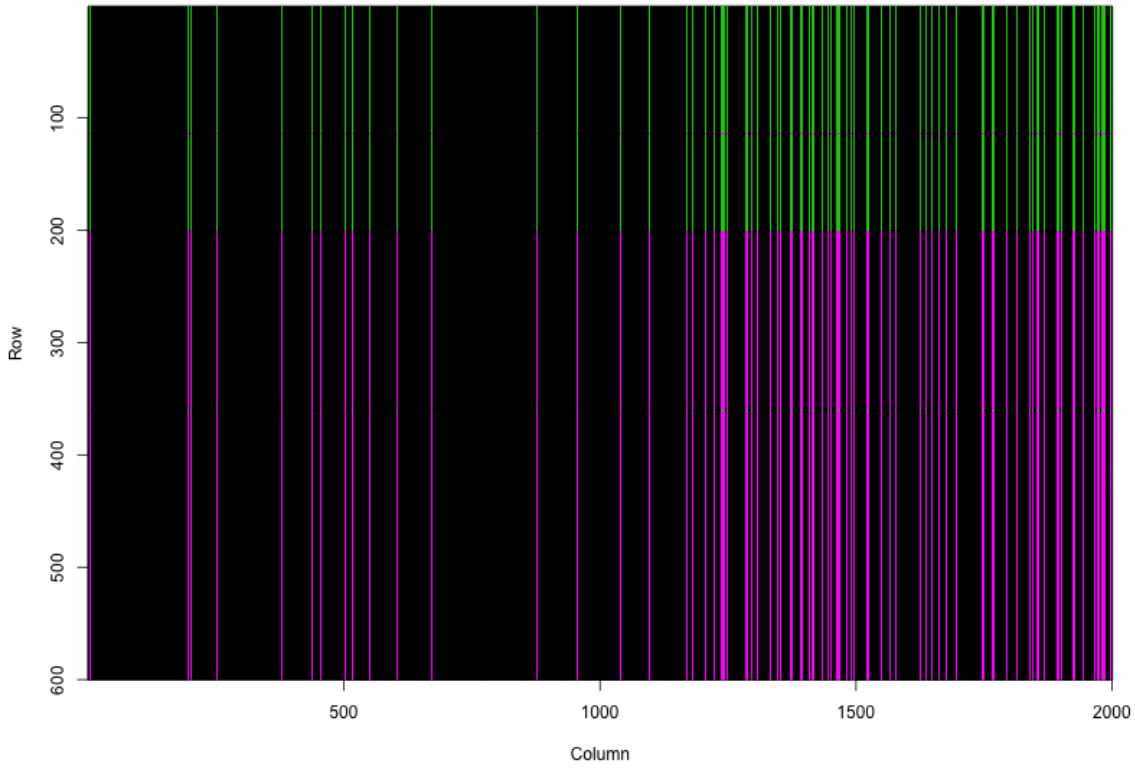

Figure 3.23: Bicluster structure learned through the MCMC algorithm. All column bicluster specific multinomial parameters are drawn from $Dirichlet(10, 10, 10)$. Jaccard Index: 0.137

Chapter 4

Bayesian Biclustering on Ordinal Data

4.1 Introduction

Ordinal data is a statistical data type we see often in real life. For example, in movie rating websites like Internet Movie Database (IMDB), people are asked to rate a movie on a scale of 1 to 5. A rating of 5 stars is better than a 4 star rating, but we do not know *how much* better it is. Ordinal data captures the ordering information in the data and presents it as discrete values. The intervals between successive scales are usually not equal. We also see this type of data in questionnaires, reviews etc. Another everyday example is the Apple App store, which asks people to rate the apps they downloaded on an ordinal scale of up to 5 stars.

As we discussed in Chapter 1, a certain type of bicluster with coherent evaluation patterns consists of orders instead of the real numerical values of the data. Taking

movie rating data, for example, we are interested in finding individuals such that their ratings are more similar to each other, over a subset of movies. After finding those *cliques*, we can build a recommendation system for suggesting movies to users.

We have not been able to find existing biclustering methods for ordinal data. We here propose two Bayesian Biclustering algorithms, one using latent variable and cutoff points from the normal distribution to model ordinal data, and the other one using a mixture of binomial and discrete uniform distributions. There are more parameters in the normal cutoff model when levels of data are higher, while the intuition behind the modeling is more straightforward. The mixture model uses a fixed number of two parameters for each bicluster and is easier in terms of implementation and converges faster. In the following sections, we will first illustrate the Bayesian modeling of Uniform Binomial Mixture model (UBM) and test the model with a simulated data set. The Normal Random Cutoff model (NRC) will be discussed later and the general framework will be presented at the end of this chapter.

4.2 Bayesian Biclustering with Uniform Binomial Mixture model (UBM)

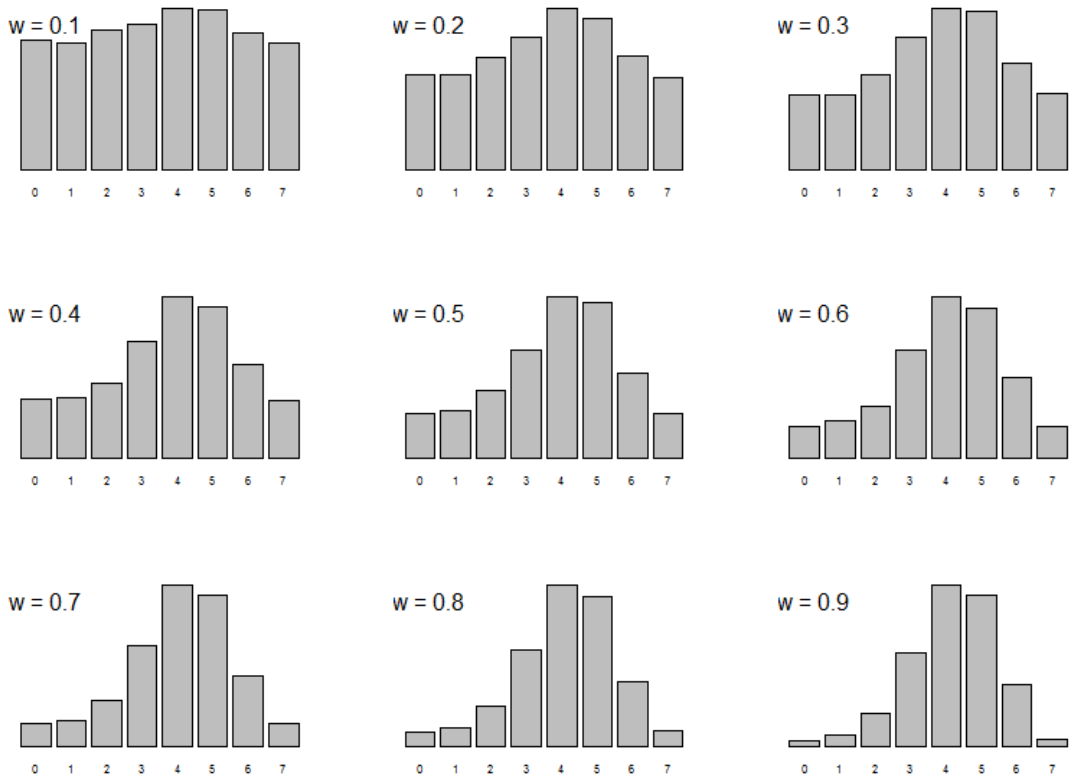

Figure 4.1: Graphical illustration of the Uniform Binomial model. In this case, the number of ordinal levels $\mathbf{M} = 8$, w is the proportion of the Binomial component in the mixture distribution. The probability parameter of the Binomial component is $p = 0.7$

We propose a new Bayesian method which uses a mixture of discrete uniform distribution and binomial distribution to approximate the ordinal distribution. Basically, we assume that each column of a given bicluster is from its cluster specific mixture distribution. This approach is very fast when dealing with ordinal data with

a high levels, **M**. Figure 4.1 illustrates 9 mixtures with different proportions of binomial components as indicated by parameter w. Notice this model can only cope with uni-modal distributions. As in reality, most ordinal distributions have a single mode and our model have a great range of applications even under this parametric setting.

4.2.1 Notations

We use a similar notation scheme as we did for the Bayesian BiClustering for Categorical model in previous chapter. $\mathbf{Y}$ is the data matrix which stores the ordinal data. Let $\mathbf{M+1}$ be the number of levels presented in the data.

Let K be the number of clusters in our algorithm, Z_i represent the cluster ID for object i and $\mathbf{S_j}$ be the column pattern indicator for the j^{th} column, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$. Z_i can take values from $\{1, 2, \cdots, K\}$, $i = 1, 2, \ldots, I$. Each object will be and can only be assigned into one cluster.

Same as before, given K clusters, for each column variable j, there are $2^K - K$ different configurations to measure the similarity among the K biclusters.

Denote w_{kj} as the proportion of the Binomial component for bicluster k in column j. Denote p_{kj} as the binomial probability parameter for bicluster k in column j. The parameters in our **UBM** model are: $\mathbf{Z}$, $\mathbf{S}$, $\mathbf{W}$, and $\mathbf{P}$.

4.2.2 Model Settings

In this model, we assume that columns are independent of each other, rows are independent conditonal on their cluster assignment. Under this assumption, we use

a mixture of Binomial and Uniform to model the ordinal data.

$$\begin{aligned} Y_{ij} &\sim w \cdot A + (1-w) \cdot B \\ A &\sim Binom(M, \mathbf{p}) \\ B &\sim \text{Discrete Uniform } on\ 0 \dots M \end{aligned} \tag{4.1}$$

where w is the binomial proportion of the mixture, p is the binomial probability parameter, and $M+1$ is the levels of the data.

The distribution of the j^{th} variable of object i, given its cluster assignment Z_i, distinction pattern indicator S_j, proportion of Binomial component $w_{Z_i,j}$ and Binomial parameter $p_{Z_i,j}$, can be expressed as follows:

$$\begin{aligned} f(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{W}, \mathbf{P}) = & \quad w_{Z_i,j} \binom{M}{y_{ij}} p_{Z_i,j}^{y_{ij}} (1-p_{Z_i,j})^{M-y_{ij}} \\ & + \quad (1-w_{Z_i,j}) \frac{1}{M+1} \end{aligned} \tag{4.2}$$

The full likelihood can be written as:

$$\begin{aligned} & P(\mathbf{Y} \mid \mathbf{Z}, \mathbf{S}, \mathbf{W}, \mathbf{P}) \\ & = \prod_{i=1}^{I} \prod_{j=1}^{J} \left\{ w_{Z_i,j} \binom{M}{y_{ij}} p_{Z_i,j}^{y_{ij}} (1-p_{Z_i,j})^{M-y_{ij}} + \frac{1-w_{Z_i,j}}{M+1} \right\} \end{aligned} \tag{4.3}$$

Priors

We model the assignment of Z_i as a Chinese Restaurant Process, and give the prior as:

$$P(\mathbf{Z}) \propto \frac{\Gamma(\alpha_z){\alpha_z}^K}{\Gamma(\alpha_z + I)} \prod_k \Gamma(C_k) \tag{4.4}$$

where C_k is the size of cluster k, α_z is the concentration parameter from the Chinese Restaurant Process, and K is the total number of clusters.

We give uniform prior for p and w. For S, we use the same penalty factor $0 < a < 1$ to penalize the inclusion of distinctive clusters.

$$\begin{aligned} p \mid Z, S &\sim Unif(0,1) \\ w \mid Z, S &\sim Unif(0,1) \\ P(S \mid Z) &\propto \prod_j \frac{a^{\sum_k S_{kj}}}{\sum_{S_j} a^{\sum_k S_{kj}}} \end{aligned}$$

Under this setting, the joint posterior can be written as

$$\begin{aligned} P(Z, P, W, S \mid Y) &\propto P(Y \mid Z, P, W, S) \cdot P(Z) \cdot P(P) \cdot P(W) \cdot P(S) \\ &\propto \prod_i \prod_j f(y_{ij} \mid Z, P, W, S) \\ &\cdot \frac{\Gamma(\alpha_z){\alpha_z}^K}{\Gamma(\alpha_z + I)} \prod_k \Gamma(C_k) \\ &\cdot \prod_j \frac{a^{\sum_k S_{kj}}}{\sum_{S_j} a^{\sum_k S_{kj}}} \end{aligned} \tag{4.5}$$

where $f(y_{ij} \mid Z, S, W, P)$ is defined as in 4.2.

4.2.3 Sampling Methods

Similarly, we use a Gibbs sampling procedure to sample **Z**, **S**, **W** and **P** iteratively by conditioning on all other parameters to obtain a sequence of samples to approximate the joint posterior distribution.

The detailed sampling procedure is as follows:

1. Sample Z

   Because the parameter space for Z_i is $\{0, 1, \cdots, K\}$, we can calculate the conditional posterior probability for Z_i of taking each of those possible values and sample Z_i from a multinomial distribution proportional to this posterior probability.

$$\begin{aligned} P(Z_i \mid Z^{-i}, P, W, S, Y) &\propto P(Y \mid Z_i, Z^{-i}, P, W, S) \cdot P(Z_i \mid Z^{-i}) \\ &\propto \prod_{j=1}^{J} f(y_{ij} \mid Z, S, W, P) \cdot P(Z_i \mid Z^{-i}) \end{aligned} \tag{4.6}$$

   where $f(y_{ij} \mid Z, S, W, P)$ is defined as in 4.2.

   Conditional on all other Z_is, the conditional prior for Z_i becomes

$$P(Z_i = k \mid Z^{-i}) = \begin{cases} \frac{\alpha_z}{\alpha_z + I - 1} & \text{if k is a new cluster} \\ \frac{C_k^{-i}}{\alpha_z + I - 1} & \text{if k is an existing cluster} \end{cases} \tag{4.7}$$

   For a new cluster, p and w are drawn directly from the prior uniform distributions; S is randomly assign with 0s and 1s.

2. Sample P, W, S

Unlike the Bayesian BiCluster for Categorical data model, we cannot integrate out P, W from the model and sample S directly. It is difficult to sample directly from the mixture distribution because it cannot be derived to a simple form for sampling. We thus reparametrize the model and use Multiple-try Metropolis (Liu (2008), Liu et al. (2000)) to sample P, W, S.

Considering a single column in our model, we drop the j index from our model for simple illustration

$$let\ \theta = (p, w)$$

$$r = \begin{cases} S = (S_1, S_2, \ldots, S_K) \\ \theta = (\theta_0, \Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_K) \end{cases} \tag{4.8}$$

$\Theta = (P, W)$ can be restructured from r as

$$\begin{aligned} \theta_1 = (p_1, w_1) &= \theta_0 + S_1 \cdot \Delta\theta_1 \\ \theta_2 = (p_2, w_2) &= \theta_0 + S_2 \cdot \Delta\theta_2 \\ &\ldots \\ \theta_K = (p_K, w_K) &= \theta_0 + S_K \cdot \Delta\theta_K \end{aligned} \tag{4.9}$$

Let $r^{(t)}$ be the set of parameters at time t, we define

$$\omega(r^{(1)}, r^{(2)}) = \pi(r^{(1)}) \cdot T(r^{(1)}, r^{(2)}) \tag{4.10}$$

where $\pi(r^{(t)})$ is the joint posterior for P, W, S at time t

$$\begin{aligned} \pi(r^{(t)}) &= P(S^{(t)}, \theta^{(t)} \mid \mathbf{y}) \\ &\propto P(\mathbf{y} \mid S^{(t)}, \theta^{(t)}) \cdot P(\theta^{(t)}) \cdot P(S^{(t)}) \\ P(\mathbf{y} \mid S^{(t)}, \theta^{(t)}) &\propto \prod_{k=1}^{K} f(\underset{i:Z_i=k}{y_i} \mid \theta_0^{(t)} + S_k^{(t)} \cdot \Delta\theta_k^{(t)}) \end{aligned}$$

$T(r^{(1)}, r^{(2)})$ is the jumping function, which we will explain in more details after describing the sampling procedure

$$T(r^{(1)}, r^{(2)}) = T(p^{(1)}, p^{(2)}) \cdot T(w^{(1)}, w^{(2)}) \cdot T(S^{(1)}, S^{(2)})$$

Starting from $r^{(t)}$, we draw N independent trial proposals as $r^{(t+1),1}$, $r^{(t+1),2}$, $\ldots$, $r^{(t+1),N}$, from $T(r^{(t)}, .)$. Compute $\omega(r^{(t+1),n})$ for n = 1, 2, $\ldots$, N.

Select $r^{(t+1),0}$ from the trial set $\{r^{(t+1),1}, r^{(t+1),2}, \ldots, r^{(t+1),N}\}$ with probability proportional to $\omega(r^{(t+1),n}, r^{(t)})$. Then draw $x_1^*, x_2^*, \ldots, x_{N-1}^*$ as a reference set from $T(r^{(t+1),0}, .)$. Set $x_N^* = r^{(t)}$.

Accept $r^{(t+1),0}$ with probability

$$r_g = min\left\{1, \frac{\sum_{n=1}^{N} \omega(r^{(t+1),0}, r^{(t)})}{\sum_{n=1}^{N} \omega(x_n^*, r^{(t+1),0})}\right\} \tag{4.11}$$

To jump from $p^{(t)}$ to $p^{(t+1)}$, we first map p to the real axis by doing a logit

transformation and then draw

$$\epsilon_p \sim N(0, \sigma_p^2)$$

$$log\frac{p^{(t+1)}}{1-p^{(t+1)}} = \epsilon_p + log\frac{p^{(t)}}{1-p^{(t)}}$$

the proposal jumping function is

$$T(p^{(t)}, p^{(t+1)}) \propto N(\epsilon_p; 0, \sigma_p^2) \cdot (\frac{1}{p^{(t)}} + \frac{1}{1-p^{(t)}})$$

Similarly, for w, we have

$$\epsilon_w \sim N(0, \sigma_w^2)$$

$$log\frac{w^{(t+1)}}{1-w^{(t+1)}} = \epsilon_w + log\frac{w^{(t)}}{1-w^{(t)}}$$

the proposal jumping function is

$$T(w^{(t)}, w^{(t+1)}) \propto N(\epsilon_w; 0, \sigma_w^2) \cdot (\frac{1}{w^{(t)}} + \frac{1}{1-w^{(t)}})$$

To propose $S^{(t+1)}$ based on $S^{(t)}$, we let p_s be the Bernouli probability that the element of S would change value, e.g. from 0 to 1, or from 1 to 0. Let n_s be the number of elements that changed values during the proposal, we have

$$T(S^{(t)}, S^{(t+1)}) = p_s^{n_s} \cdot (1-p_s)^{K-n_s}$$

We iterate between sampling Z and P, W, S until the algorithm converges.

4.2.4 Determination of Number of Clusters

The number of clusters K is incorporated into our model. The joint posterior probabilities for different numbers of clusters are up to a normalizing constant and are thus comparable. We can compare the joint posterior to determine the optimal number of clusters.

We will run independent chains starting with different number of clusters and compare the joint posterior modes for different numbers of clusters in determining the optimal number of clusters.

4.2.5 Algorithm Summary

1. Start with $K = 2$.
2. Arbitrarily set values of $Z^{(1)}$, $P^{(1)}$, $W^{(1)}$ and $S^{(1)}$ in the Gibbs algorithm.
3. Suppose we have already obtained $Z^{(t)}$, $P^{(1)}$, $W^{(1)}$ and $S^{(1)}$, update Z, P, W, S by taking the following steps:
   (a) Sample $Z^{(t+1)}$ from Y, $P^{(t)}$, $W^{(t)}$, $S^{(t)}$, and $Z^{(t)}$.
   (b) Sample $P^{(t+1)}$, $W^{(t+1)}$ and $S^{(t+1)}$ from Y, $Z^{(t+1)}$, $P(t)$, $W^{(t)}$ and $S^{(t)}$.
   (c) $t \to t + 1$.
4. Iteratively run Step 3 until convergence.
5. Calculate the joint posterior mode for the final sample and record it.
6. $K \to K + 1$ if K does not exceed the pre-specified upper bound. Return to Step 2.

7. Select the value of K that leads to the highest joint posterior mode as the optimal number of clusters. Output the posterior samples of Z, P, W and S under this K.

4.2.6 Simulation Study

Date Generation

In this simulation study, we generated a 600 by 2000 matrix for 3 biclusters as plotted in Figure 4.2, and the number of levels of the ordinal data was set to 5. Column-wise, the 3 bicluster foregrounds were drawn from $(p, w) = (0.2, 0.9), (0.6, 0.9), (0.9, 0.9)$ respectively, and the background samples were drawn from $(p, w) = (0.5, 0.1)$. Notice that in this data set, we embed all $2^3 - 3 = 5$ different distinction patterns into the data. The simulated bicluster structure is presented in Figure 4.2.

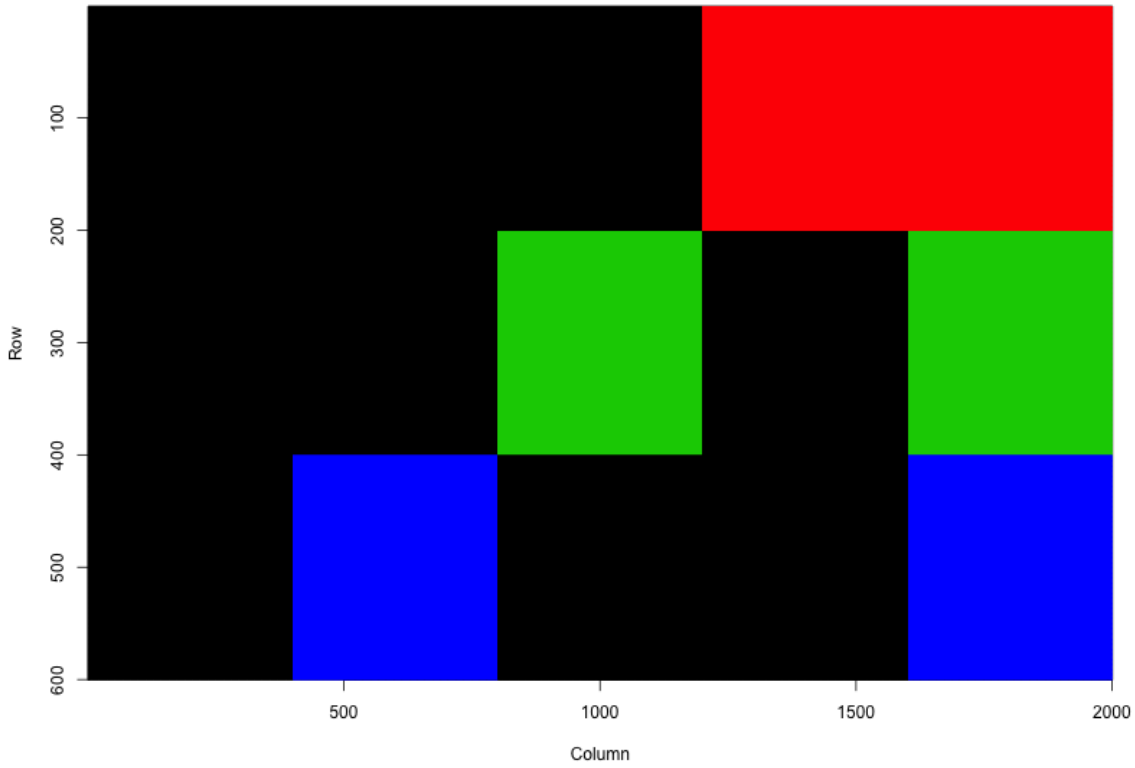

Figure 4.2: The true bicluster structure from which the data is simulated. Each color demonstrates a different bicluster. 3 biclusters embeded in this simulation

We followed the sampling procedure presented in section 4.2.5. The number of multiple-try was set to 1,000. For each number of biclusters $\mathbf{K}$, we ran the MCMC algorithm for 1,000 steps and use the first 500 steps as burn-in. The joint posterior mode was extracted using the samples from the last 500 steps. We tested different values of $\mathbf{K}$, $K = 2, 3, \ldots, 8$, and found $K = 3$ maximizes the joint posterior, which is consistent with truth in the simulation data.

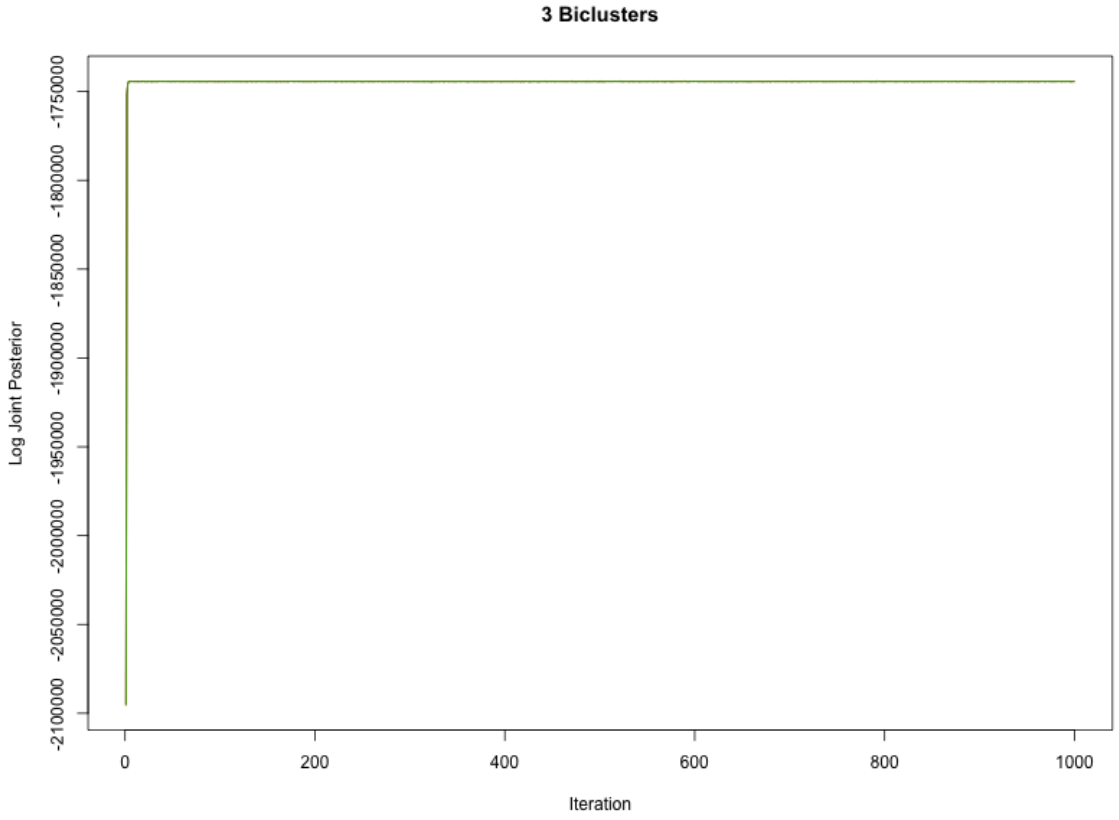

Figure 4.3: Trace plot for the joint posterior of three parallel chains.

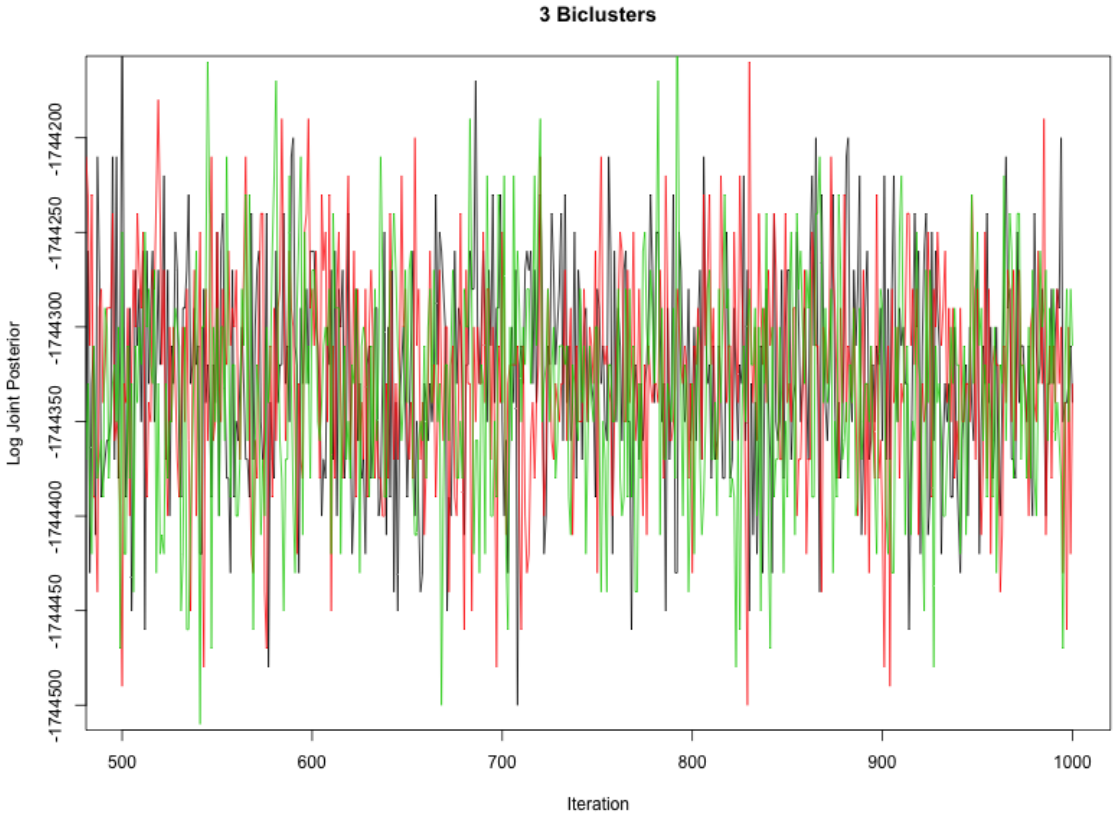

Figure 4.4: Trace plot for the joint posterior of three parallel chains.

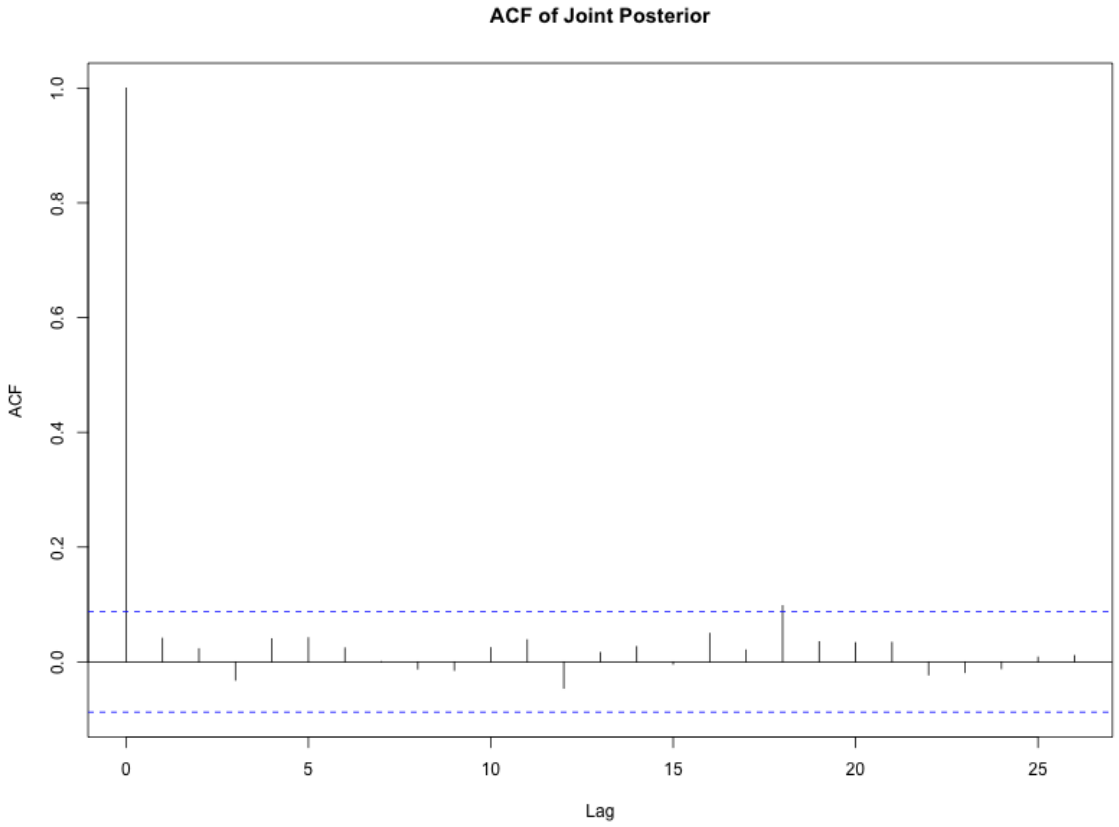

Figure 4.5: Autocorrelation plot for the joint posterior.

We ran 3 parallel MCMC chains with different starting values for P, W and S. The initial values for Z were using the results from hierarchical clustering by treating data as continuous and used Euclidean distance as the measure. The diagnostics for the parallel chains are presented in Figure 4.3 and Figure 4.5. The recovered bicluster structure is presented in Figure 4.6.

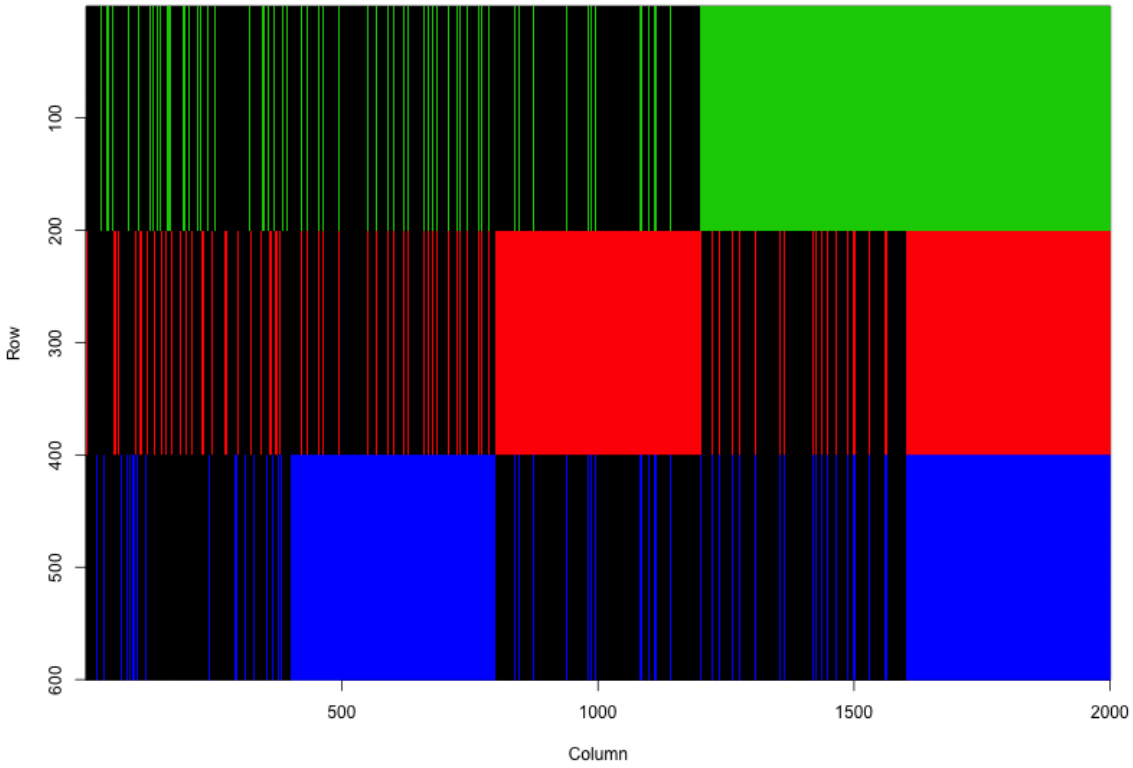

Figure 4.6: the biclusters learned through the MCMC algorithm. They resemble the true structure in general except for a small amount of errors. Jaccard Index: 0.84

4.3 Modeling Ordinal Data with Normal Random Cutoff model (NRC)

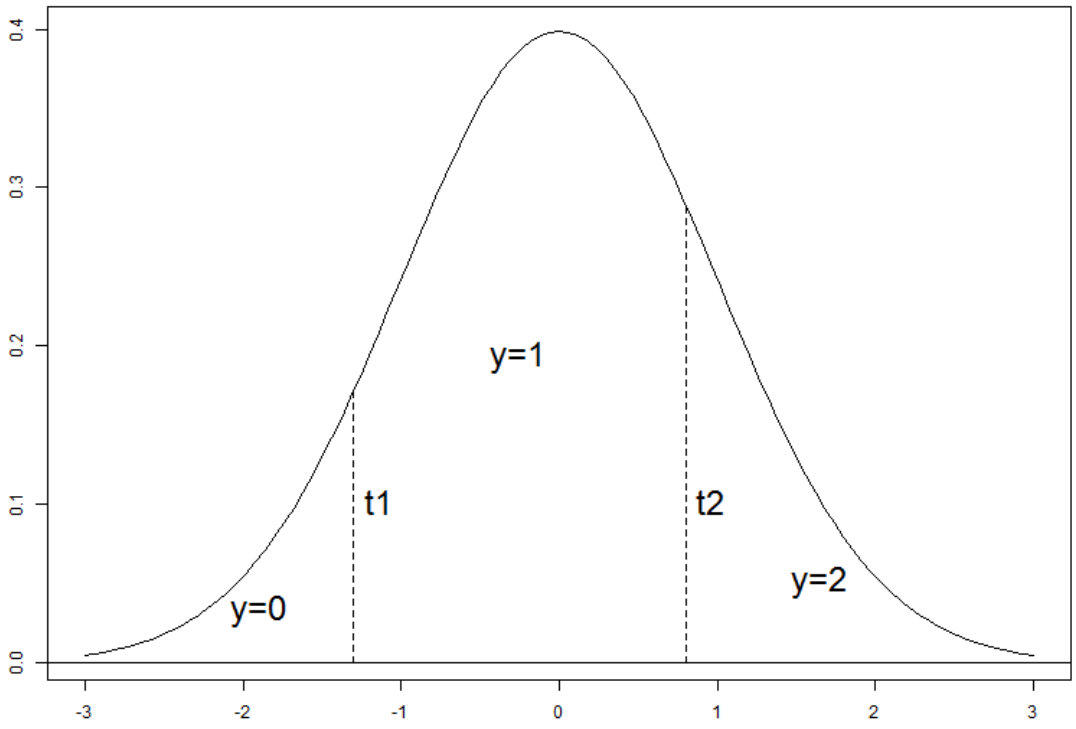

Figure 4.7: Graphical illustration of the Normal Random Cutoff model. In this case, the number of ordinal levels $\mathbf{M} = 3$, the 2 cutoffs are t_1 and t_2. When latent variable falls to the left of t_1, the ordinal variable is 0; when the latent variable falls between t_1 and t_2, the ordinal variable y is 1; when it falls to the right of t_2, y is 2.

Besides modeling discrete ordinal data using a mixture of Binomial and Discrete Uniform distributions, there are a few alternatives. Because ordinal data contains only the order information, a natural way of modeling this type of data is to use a continuous latent variable that comes from a standard normal distribution, as in

Figure 4.7. We create **M-1** random cutoffs to divide the distribution into **M** disjoint intervals. The order is then determined by which interval the latent variable falls into. Suppose there are 3 levels in the ordinal data set and the cutoffs on the standard normal distribution are -1, 1, a latent variable whose value is less than -1 will be mapped to 0 in the ordinal data set; latent variables that fall into the interval $[-1, 1]$ will be mapped to 1, and latent variables greater than 1 will be mapped to 2. Under this Normal Random Cutoff model setting, an ordinal distribution can be represented using a cutoff vector on a standard normal distribution.

Let K be the number of clusters in our algorithm, Z_i represent the cluster ID for object i, and $\mathbf{S_j}$ be the column pattern indicator for the j^{th} column, $j = 1, 2, \ldots, J$. Z_i can take values from $\{1, 2, \cdots, K\}$, $i = 1, 2, \ldots, I$. Each object will be and can only be assigned into one cluster.

Same as before, given K clusters, for each column variable j, there are $2^K - K$ different configurations to measure the similarity among the K biclusters.

Let $\mathbf{L}$ be aa I by J matrix, in which each cell stores the continuous latent variable l_{ij}. We use $\mathbf{T}$ to denote the normal cutoff matrix, with each entry representing the cutoff vector $(t_{j,S_j,Z_i,1}, t_{j,S_j,Z_i,2})$, which is the corresponding lower and upper cutoff for the standard normal distribution.

The parameters in the model are Z, S, and T.

As we know from our previous model for ordinal data, Multiple-try Metropolis (MTM) is computationally very expensive. Instead of using MTM, we propose an approximation for the Normal Random Cutoff model.

4.3.1 Model Settings

In this model, we assume that columns are independent of each other, rows are independent conditonal on their cluster assignment. We assume The distribution of the j^{th} variable of object i, given its cluster ID Z_i, distinction pattern indicator S_j and normal cutoff parameter $\mathbf{T}$ can be expressed as follows:

$$P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T}) = \begin{cases} \Phi(t_{j,S_j,Z_i,1}) & \text{if } y_{ij} = 0 \\ \Phi(t_{j,S_j,Z_i,2}) - \Phi(t_{j,S_j,Z_i,1}) & \text{if } y_{ij} = 1 \\ \Phi(t_{j,S_j,Z_i,2}) & \text{if } y_{ij} = 2 \end{cases} \tag{4.12}$$

where $\Phi(.)$ is the cumulative density function of the standard normal distribution.

The full likelihood of this model is derived as follows:

$$P(Y \mid \mathbf{Z}, \mathbf{S}, \mathbf{T}) = \prod_{i=1}^{I} \prod_{j=1}^{J} P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T}) \tag{4.13}$$

where $P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T})$ is defined as in 4.12.

Priors

We give $\mathbf{Z}$ an independent joint uniform prior on $\{1, \cdots, K\}$.

$$P\Big(\mathbf{Z} = (k_1, k_2, \ldots, k_I)\Big) = (\frac{1}{K})^I \tag{4.14}$$

where $k_1, k_2, \ldots, k_I \in \{1, 2, \ldots, K\}$

The priors for $\mathbf{S}$ was given to penalize the inclusion of distinctive clusters. Let n_j

be the number of 1s in S_j. We let $P(S_j \mid Z) \propto a^{n_j}$, where $a < 1$ is a positive number.

$$P(S \mid Z) \propto \prod_j a^{n_j} \tag{4.15}$$

We give priors for each $t_{j,S_j,Z_i,1}$ and $t_{j,S_j,Z_i,2}$ such that they are order statistics of two independent uniform variables on the interval [-5, 5]. The interval is set this way on one hand to cover the main part of a standard normal distribution, and on the other hand, to avoid the occurrence of extreme values for $\mathbf{T}$:

$$\begin{aligned} t_{j,S_j,Z_i,1} \mid Z, S &\sim unif(-5, 5) \\ t_{j,S_j,Z_i,2} \mid Z, S &\sim unif(-5, 5) \end{aligned} \tag{4.16}$$

4.3.2 Sampling Methods

We constructe a Gibbs sampling procedure to sample $\mathbf{Z}$, $\mathbf{S}$ and $\mathbf{T}$. For each parameter, we iteratively update it by conditioning on all other parameters to obtain a sequence of samples to approximate the joint posterior distribution.

The detailed sampling procedure is as follows:

1. Sample Z given Y, S, T

   The parameter space for Z_i is $\{1, \cdots, K\}$, and we can calculate the conditional posterior probability for Z_i of taking each of those possible values:

$$P(Z_i = k \mid \mathbf{Z}^{-i}, \mathbf{Y}, \mathbf{S}, \mathbf{T}) \propto \prod_{j=1}^{J} P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T}) \tag{4.17}$$

$P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T})$ is defined in 4.12. We then sample Z_i from a multinomial distribution proportional to this quantity.

2. Sample T given Y, S, Z

   If $S_{kj} = 1$, then cluster k is a distinctive cluster, we will sample $T_{j,S_j,k,1}$ and $T_{j,S_j,k,2}$ based on the data only belongs to cluster k.

   If $S_{kj} = 0$, then cluster k is one of the few identical clusters, we will sample $T_{j,S_j,k,1}$ and $T_{j,S_j,k,2}$ based on the combined data that belongs to those clusters; This is done only once for each of the identical clusters and the same sampled value of $T_{j,S_j,k,1}$ and $T_{j,S_j,k,2}$ are assigned to each of them.

   To sample $\mathbf{T}$, we will first sample latent variable $\mathbf{L}$ according to:

$$l_{i,j,S_j,Z_i} \sim \begin{cases} Truncated - Normal(0,1) \text{ on } (-5, t_{j,S_j,Z_i,1}) & \text{if } y_{ij} = 0 \\ Truncated - Normal(0,1) \text{ on } (t_{j,S_j,Z_i,1}, t_{j,S_j,Z_i,2}) & \text{if } y_{ij} = 1 \\ Truncated - Normal(0,1) \text{ on } (t_{j,S_j,Z_i,2}, 5) & \text{if } y_{ij} = 2 \end{cases} \tag{4.18}$$

   After obtaining the samples for $\mathbf{L}$, we sample each individual $t_{j,S_j,Z_i,1}$ and $t_{j,S_j,Z_i,2}$ as follows:

$$\begin{aligned} t_{j,S_j,Z_i,1} &\sim Unif(\underset{y_{ij}=0}{max}\{l_{i,j,S_j,Z_i}\}, \underset{y_{ij}=1}{min}\{l_{i,j,S_j,Z_i}\}) \\ t_{j,S_j,Z_i,2} &\sim Unif(\underset{y_{ij}=0}{max}\{l_{i,j,S_j,Z_i}\}, \underset{y_{ij}=1}{min}\{l_{i,j,S_j,Z_i}\}) \end{aligned} \tag{4.19}$$

   The above two steps (4.18 and 4.19) are repeated for many steps to obtain a good sample of $\mathbf{T}$.

There are potentially some boundary problems when some levels have zero counts. In this situation, we can sample $\mathbf{t}$ using the nearest known border values as the ends of the intervals.

We do this for every possible distinction pattern of S_j and for every column j and obtain a $\mathbf{T}$ matrix of K by J by $(2^K - K)$ by 2.

3. Sample S given Y, T, Z

   There are $2^K - K$ different configurations for S_j. We can thus calculate the posterior probability for each S_j and then draw S_j proportional to

$$P(S_j = m \mid \mathbf{Y}, \mathbf{T}, \mathbf{Z}, \mathbf{S}_{[-j]}) \propto \prod_{k=1}^{K} \prod_{\{i:Z_i=k\}} P(y_{ij} \mid \mathbf{Z}, \mathbf{S}, \mathbf{T}) \tag{4.20}$$

4.3.3 Determination of Number of Clusters

We can use Bayesian Information Criterion (BIC) (Schwarz (1978)) to determine the number of biclusters in the data. BIC is calculated by the following equation:

$$BIC = -2 \log Lik + m \log n$$

where Lik represents the likelihood, m the number of free parameters and n the data size.

Chapter 5

Inference of Human Population Structure Using HapMap Data

People are different. However, any two individuals in the world share more than 99% of their DNA in common. It is the less than 1% difference that makes each person unique. Among the many forms of DNA variations, single-nucleotide polymorphism (SNP) is the most common type, which is defined as the DNA sequence variation of a single nucleotide between members of biological species or homologous chromosomes in a human. The international HapMap project has made available the SNP data of thousands of individuals across the world. We describe an application of the model-based Biclustering method developed in Chapter 3 to infer the similarities and differences between human populations using multilocus genotype data . In contrast to existing methods, our method can locate SNPs that are specific to given subpopulation groups. We show that the method can produce highly accurate classification of populations using individual genotype data and locate the differences

between population groups. The Biclustering process can be used as a variable selection step prior to existing population inference procedures. The algorithm can also provide insight to the genome-wide association study (GWAS) by finding SNPs that are common to different ethnic groups.

5.1 Terminology

SNPs are DNA sequence variations of single nucleotides between individuals or biological species, which occur more frequently in the non-coding regions of the genome than in coding regions.

An *allele* is one of a number of alternative forms of a single gene or genetic *locus* (Malats and Calafell (2003)). Most SNPs have two alleles because it is highly unlikely that the same mutation would occur on the same nucelotide more than once during the production of germ cells, given the large number of nucleotides in the human genome. In the following analysis of the Human HapMap Project SNPs, we focus only on SNPs that have two alleles. For *diploid* organisms, there are two sets of chromosomes in the genome, which are called *homologous chromosomes*. Each of the two homologous chromosomes has exactly one copy of alleles on them. For example, at a given genetic site (locus), there are two variants **A** and **a**. If random mating is assumed, there are three possible combinations of alleles for this locus considering both of the homologous chromosomes, which are {**AA**}, {**Aa**} and {**aa**}. Those three combinations are called *genotypes* of the allele.

Minor allele frequency (MAF) is defined as the frequency of the less common allele in a given population.

5.2 Introduction

As of today, the world has more than 7 billion people (Nations (2013)) and everyone is unique. Differences between people can be tracked down to the DNA of human genomes, the blueprint of life. Human genomes differ in many ways. There are small variations at the nucleotide level, known as SNP, and large variations such as insertions, deletions and copy number variations (CNVs). Those variations in DNA cause difference among individuals and create a world of variety. DNA-level differences result in the variations in phenotypes, which can be manifested as differences in physical properties such as height and weight.

Over the past two decades, scientists from different countries have worked together on identifying the composition of the human genome. The completion of the human genome project has given researchers the opportunity to understand genetic diseases through association studies, mapping oncogenes and mutations linked to cancers, certain pathogens, etc. Genome-wide association study (GWAS) focuses on comparing the allele frequencies of diseased individuals with normal ones to determine whether their allele frequency variation is linked to certain disease. As we can see today in the open database of GWAS results, the study has identified thousands of SNPs associated with more than 300 diseases (Johnson and O'Donnell (2009)). Nevertheless, the case-control designs of GWAS study are susceptible to confounding from population stratification, where the allele being studied has different frequencies across subgroups of the population. Population stratification is caused by nonrandom mating due to physical separation and genetic drift of alleles. Differences in allele frequency are not necessarily caused by disease: it can also be due to ancestry differences, which makes

population stratification a confounding factor in association studies.

Population stratification can occur when performing case-control studies in a non-homogeneous population. It's critical to find ancestry background allele frequencies between cases and controls. The case-control designs are susceptile to confounding from population stratification if the allele being studied has variation across subgroups of the population. The subgroups may also be different in term of their baseline risk of the disease. One example of the problem without population stratifcation can be seen from the study from (Knowler et al. (1988)), which shows an inverse association between variants in immunoglobulin and non-insulin-dependent diabetes mellitus among Gila River Indian Community residents. However, when population stratification was performed by eliminating the Caucasian heritage factor, the inverse association disappeared.

It has been shown that many alleles have different frequencies across populations (Perez-Lezaun et al. (1997); Goddard et al. (2000)) and the extent of variation is linked to the genetic distance between these populations. In association studies, the variation of disease allele frequencies should be corrected with the baseline frequencies of different subpopulation groups before drawing valid conclusions (Caporaso et al. (1999)). The questions to be addressed are: how much variations there are in allele frequencies in a given population, and what the baselines of those populations are?

One approach to deal with confounding variables in association study is to use matched individuals with their geographical or ethnic groups. However, this method cannot handle the cases of admixture populations (populations with mixed ancestry). Part of African American populations in the United States, and *mestizo* populations

in Mexico, are examples of admixture populations (Salari et al. (2005); Parra et al. (1998)). It is very difficult to apply matching for individuals with multiple ethnic ancestors. This thus poses a great challenge to association studies with large sample size.

Population stratification can lead to false positive or false negative results in SNP association studies (Choudhry et al. (2006); Freedman et al. (2004); Barnholtz-Sloan et al. (2008); Cardon and Palmer (2003)). A few methods have been developed to address these issues. The Fixation index (Fst) (Weir and Cockerham (1984); Cockerham and Weir (1993)) is a measure used to quantify the genetic differentiation of a population due to genetic structure, and it is widely used in population genetics. The Fst calculation is based on the variance of allele frequencies between populations and it depends heavily on the number of SNPs used.

STRUCTURE (Pritchard et al. (2000); Falush et al. (2003, 2007); Hubisz et al. (2009)) used a model based clustering method and assigns individuals into subpopulations by computing the likelihood of each genotype being in each of those subpopulations. EIGENSTRAT (Price et al. (2010)) uses principal component analysis (PCA) to perform data dimension reduction while keeping most variability. Genomic controls (Devlin et al. (2004)) uses genetic markers that are not linked to the trait to adjust the inflation of the association statistics. The rescaling of the chi-square statistics is done by using a overall uniform inflation factor for all markers and may introduce problems.

PCA based methods are becoming popular in GWAS due to their small computational cost, and can be performed on a whole genome scale (Burton et al. (2007);

Craddock et al. (2010); Yeager et al. (2007); Hunter et al. (2007)). EIGENSTRAT is one of the most widely used PCA methods, though there are many alternative extensions that are also based on PCA (Epstein et al. (2007); Serre et al. (2008); Li and Yu (2008); Jombart et al. (2010)). Due to computational cost, some studies will not perform the analysis on the whole genome set, and instead only use ancestry-informative markers (AIMs), a subset of markers that show greater differences between ancestral populations (Bauchet et al. (2007); Tian et al. (2008); Seldin and Price (2008); Mao et al. (2007); Tian et al. (2006)). However, with populations of unknown origins or admixture populations, existing AIMs panels are of little use (Barnholtz-Sloan et al. (2008)). PLINK (Purcell et al. (2007)) also provides a toolset for genetic association analysis. With a predefined number of clusters, the CochranMantelHaenszel test (CMH) in PLINK can be used to test overall disease and gene association. Salvi et al. (2011) applied aforementioned methods in two real datasets and compared their performance.

In this Chapter, we propose a Bayesian BiClustering model for population stratification and background allele frequency inference. Our algorithm not only detects the differences in the genetic variations between individuals and separates them into subpopulations, but also estimates the background allele frequencies for each SNP. Furthermore, we can identify SNPs that are common to all populations, as well as those that are specific to subpopulation groups. The number of shared SNPs between two subpopulation groups can also be used as an indicator of the genetic closeness between them.

5.2.1 Population Structure Inference

Population structure inference is the problem of assigning individuals to different clusters according to their differences in allele frequencies between subpopulations due to different ancestry. Population structure is also called population stratification in population genetics. The major cause of population stratification is non-random mating between groups of individuals. This can be attributed to physical separation such as geographical isolation.

Genetic markers (e.g SNPs, RFLP etc.) are good measures for inferring the hidden population structures in a group of individuals (Pritchard et al. (2000)). Many algorithms have been developed to solve this problem. Suppose we have individuals' genetic information, and we want to classify them into different groups. There are in general two major types of clustering methods: distance based clustering and model based clustering. In the case of distance based clustering, one needs to define a distance measure for the pairwise genetic distance between individuals. Traditional clustering algorithms can then be applied once the distance matrix is computed. For model based clustering, it is assumed that every individual is drawn from a cluster specific parametric model. Distance based methods are straightforward, and there are many classical methods available, such as *k-means*. Disadvantages of distance based clustering include: (1) it depends on the choice of distance measure; (2) it is difficult to interpret the statistical meaning of such classifications; (3) it cannot quantify uncertainties. One popular algorithm based on distance measure is EIGENSTRAT proposed by Price et al. (2006). It first uses Principle Component Analysis (PCA) Jackson (2005) to determine the directions of the genetic variation. After the data

dimensionality reduction, a multilinear regression is carried out using the remaining dimensions on cases and controls. The detection of ancestry differences is performed by computing the association statistics.

Another widely used statistical algorithm for making population structure inference is STRUCTURE (Pritchard et al. (2000), Falush et al. (2003), Falush et al. (2007)). STRUCTURE proposes two models: Model without admixture and Model with admixture. In the model without admixture, it is assumed that there are **K** populations and each individual is from one of the populations. The population here is characterized by an allele frequency matrix **P**. i.e. within a given population, each SNP site has its population specific alle frequencies for all alleles on the same site. For the model with admixture, each individual is assumed to originate from a mixture of the **K** predefined populations. There is a **K** dimensional **q** vector for each individual, which indicates the fractional heritage from each of the populations. q_{ik} is the probability that individual i comes from population k, or equivalently the percentage of genome of individual i that was derived from population k. Just as with the model without admixture, the population here can be described using an allele frequency matrix **P**.

Unlike STRUCTURE, we treat each individual as an observation from one of the unknown populations. This is more practical from the perspective of the Genome-Wide Association Study (GWAS). We are more interested in knowing the differences between those unknown populations and what caused the population differences. The admixture information is not of our primary research interest because knowing the admixture information does not directly help identity the genome sites for association

study. Under this setting, the unknown populations can be interpreted as *new human types*, which share same genomic similarity. STRUCTURE uses the allele frequency matrix to characterize each ancestral population. In our model, we use directly the genotype frequency matrix to describe our cluster. We will detail the model settings in the following section.

The primary goals of our study are

1. Grouping individuals into population groups based on their genetic differences;

2. Finding out where the genetic differences are;

3. Investigating how various disease-linked SNPs interplay with race-specific SNPs.

5.3 Data description and preprocessing

5.3.1 HapMap Project

In our study, we used the Phase III data from the international HapMap Project (Thorisson et al. (2005); Altshuler et al. (2010)). The international HapMap Project aims to construct a haplotype map of the human genome to describe common patterns of human genetic variation. It is a collaboration among organizations from Canada, China, Japan, Nigeria, the United Kingdom and the United States. The results of Phase I were published in 2005. The Phase II dataset was published in 2007, and Phase III was released in 2009.

The HapMap project focuses on common SNPs that have a minor allele frequency (MAF) greater than 5%, while HapMap project phase III managed to include SNPs

with low frequencies (MAF $< 5\%$), and thus brought the information database to an even higher resolution. The dataset from the HapMap project are the genotypes of SNPs from the sampled individuals. The alleles of nearby SNPs on a chromosome are correlated because of evolutional recombination events, which is referred as Linkage Disequilibrium (LD) (Reich et al. (2001)). Tag SNPs are the representative SNPs in a certain region for correlated SNPs (Halperin et al. (2005)). In GWAS study, to find the genetic factors that affect certain phenotypes (diseases etc.), researchers focus on those tag SNPs and identify the distribution of them across individuals with different phenotypes.

HapMap phase III collected the genotype data of 1,397 individuals from 11 populations. The data were obtained by merging the results from Affymetrix Human SNP array 6.0 and Illumine Human1M-single beadchip. After filtering out low-quality/incomplete data and post processing, the consensus genotype set was left with 1,440,616 SNPs that are polymorphic in the sampled individuals (Altshuler et al. (2010)). Some of the genetic variations are specific to populations, while some of them are specific to a set of populations. We are interested in identifying those specific SNPs patterns that define the population structures.

5.3.2 Data Preprocessing

The 11 populations are: individuals from the Centre d'Etude du Polymorphisme Humain collected in Utah, USA, with ancestry from northern and western Europe (CEU); Han Chinese in Beijing, China (CHB); Japanese in Tokyo, Japan (JPT); Yoruba in Ibadan, Nigeria (YRI); African ancestry in the southwestern USA (ASW);

Chinese in metropolitan Denver, Colorado, USA (CHD); Gujarati Indians in Houston, Texas, USA (GIH); Luhya in Webuye, Kenya (LWK); Maasai in Kinyawa, Kenya (MKK); people with Mexican ancestry in Los Angeles, California, USA (MXL); and Tuscans in Italy (Toscani in Italia, TSI).

The raw genotype data was downloaded in PLINK format (Purcell et al. (2007)) from HapMap phase III website. Due to linkage disequilibrium, we used PLINK to remove correlated SNPs in the decorrelation step by setting the threshold of r^2 to be 10^{-6}, where r^2 refers to the squared Pearson correlation coefficient of observed genotypes between SNPs. The sliding window is set to be 200 with an increasing step of 20. Also, we removed the child from each of trio-family so as to collect only unrelated individuals. This generates a 1,198 by 4,217 data matrix. The SNPs in our data are all biallelic thus the genotypes are either 0, 1 or 2. After the decorrelation step, the Minor Allele Frequencies of the 4,217 SNPs are plotted in Figure 5.1.

The short code and number of individuals for each sample populations are listed in Table 5.1

Table 5.1: All 11 populations

Short Code	Description of group	Individual numbers
ASW	African ancestry in Southwest USA	1 - 53
LWK	Luhya in Webuye, Kenya	54 - 163
MKK	Maasai in Kinyawa, Kenya	164 - 319
YRI	Yoruba in Ibadan, Nigeria	320 - 466
GIH	Gujarati Indians in Houston, Texas	467 - 567
MEX	Mexican ancestry in Los Angeles, California	568 - 625
CHB	Han Chinese in Beijing, China	626 - 762
CHD	Chinese in Metropolitan Denver, Colorado	763 - 871
JPT	Japanese in Tokyo, Japan	872 - 984
CEU	Utah residents with NW European ancestry	985 - 1096
TSI	Toscans in Italy	1097 - 1198

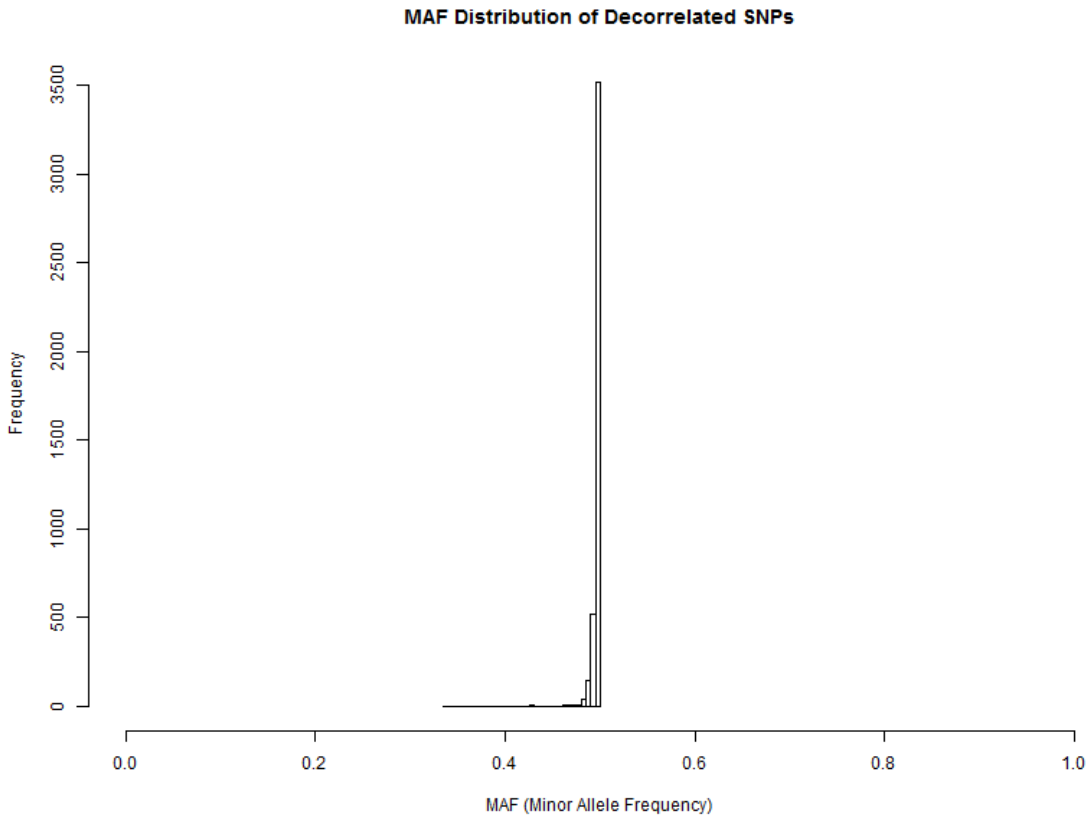

Figure 5.1: After the decorrelation step, we reduced the number of SNPs from 1.4 million to 4,217 by setting r^2 to 10^{-6} using PLINK. We calculated the MAF based on the genotype data for each of the 4,217 SNPs. The MAF ranges from 0.337 to 0.500 and are all common SNPs. This histogram shows the distribution of the MAFs in our dataset.

Our processed data is now a matrix of categorical data, taking values in three different genotypes. Each row represents an individual while each column represents a SNP. The individuals are unrelated and the SNPs are uncorrelated. We are interested in inferring the population structure in the data without using the self-reported race information, and in the mean time finding SNPs that are specific to each population from an unsupervised standpoint.

5.4 Data Analysis

We used the Bayesian Biclustering model for Categorical Data (BBCD) we developed in Chapter 3 to analyze the genotype data of 1,198 unrelated individuals on 4,217 SNPs. In the current context, there are 3 categories for each SNPs, or in other words 3 different genotypes for each SNP site, which are represented using { **0**, **1**, **2** }. The unknown populations here correspond to the clusters in this model. Each individual will be and will only be assigned into one of the clusters. Within each cluster, every column of SNP genotypes is from the same multinomial distribution. The SNP sites whose genotypes are from the same multinomial distribution across all clusters will be treated as background because they do not contribute in distinguishing populations. The primary goal is to infer the number of biclusters in the data set and find biclusters that share the same multinomial distributions on selected SNPs columns, as illustrated in Figure 5.2.

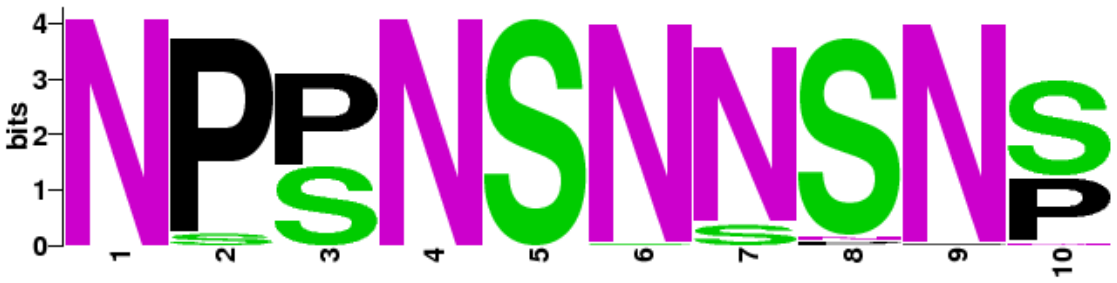

Figure 5.2: Illustration of a bicluster with three categories, {S, N, P}. The height of the respective letter is proportional to the multinomial probability of this category on the given column. The figure was generated using a modified version of Sequence Logos software Schneider and Stephens (1990)

Following the notations from Chapter 3, let K be the number of clusters in the data set, Θ be the multinomial genotype frequency matrix, M be the number of categories in the data which is 3 in this case, and S be the column similarity pattern indicator matrix. Z_i is the cluster ID for individual i, and y_{ij} is the genotype of individual i at SNP j. Given $Z_i = k$, the genotype y_{ij} is from a multinomial distribution with frequency parameter $\vec{\theta}_{kj}$.

$$y_{ij} \mid \mathbf{Z}, \mathbf{S}, \Theta \sim Multinom(\vec{\theta}_{Z_i,j})$$

The Dirichlet prior for Θ is set to be:

$$\alpha_\theta = \{2, 2, 2\}$$

and the Chinese Restaurant processs prior for Z is set to be:

$$\alpha_Z = \{1, 1, 1\}$$

The prior for $\mathbf{S}$ was given to penalize the inclusion of distinctive clusters:

$$P(S_j) \propto a^{\sum_k S_{kj}}$$

where $a = 0.1$.

We ran the algorithm starting with different number of clusters, until the number of clusters converges. The optimal number of clusters is determined by comparing the joint posterior modes of each setting.

5.4.1 Results and Biological Implications

We used all 4,217 SNPs for the 1,198 individuals to find biclusters and found 5 biclusters according to the joint posterior modes. The publicly available potential etiologic and functional association loci for human diseases and traits database was used to match the SNPs (Hindorff et al. (2009), Hindorff et al. (2011)). We further found that several diseases could potentially have different degrees of influences on the subpopulations we discovered.

We started our algorithm with 2 clusters and gradually increase the number of clusters to 3, 4, 5, 6, and 7. The number of clusters converges to 5 for any run starting with a number greater than 5 as plotted in Figure 5.3. The comparision of their respective joint posterior mode is presented in Figure 5.4.

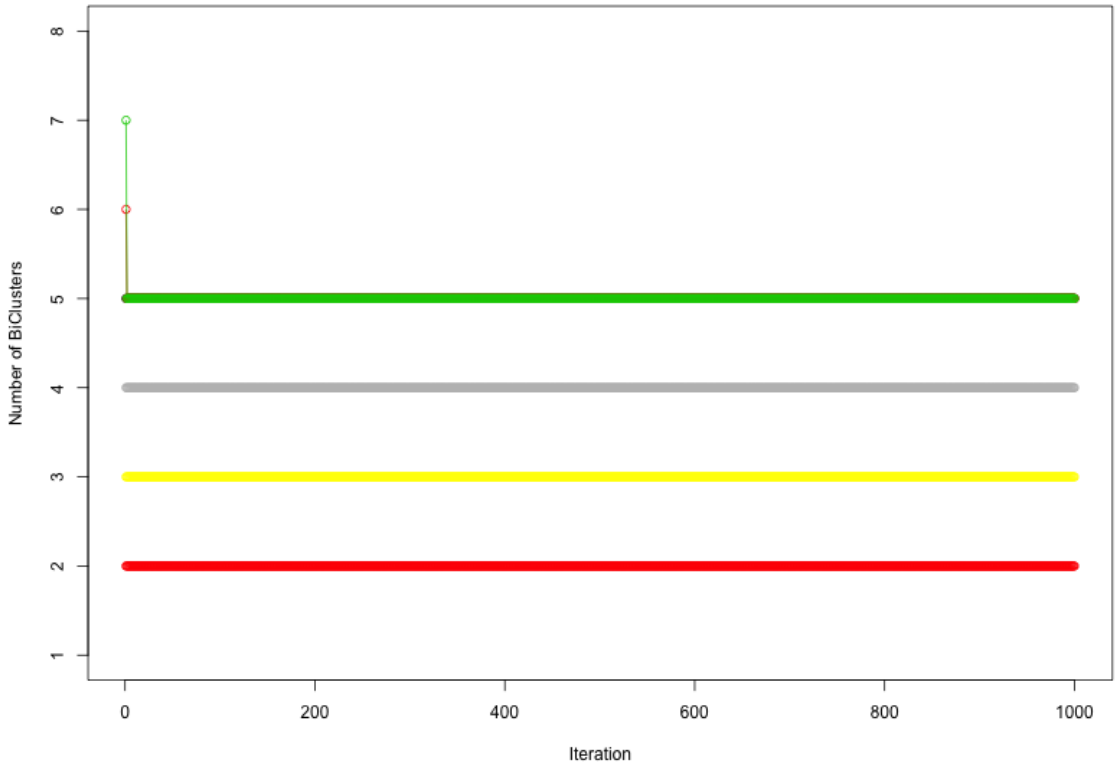

Figure 5.3: Number of clusters converges to 5 when starting with a number greater than 5.

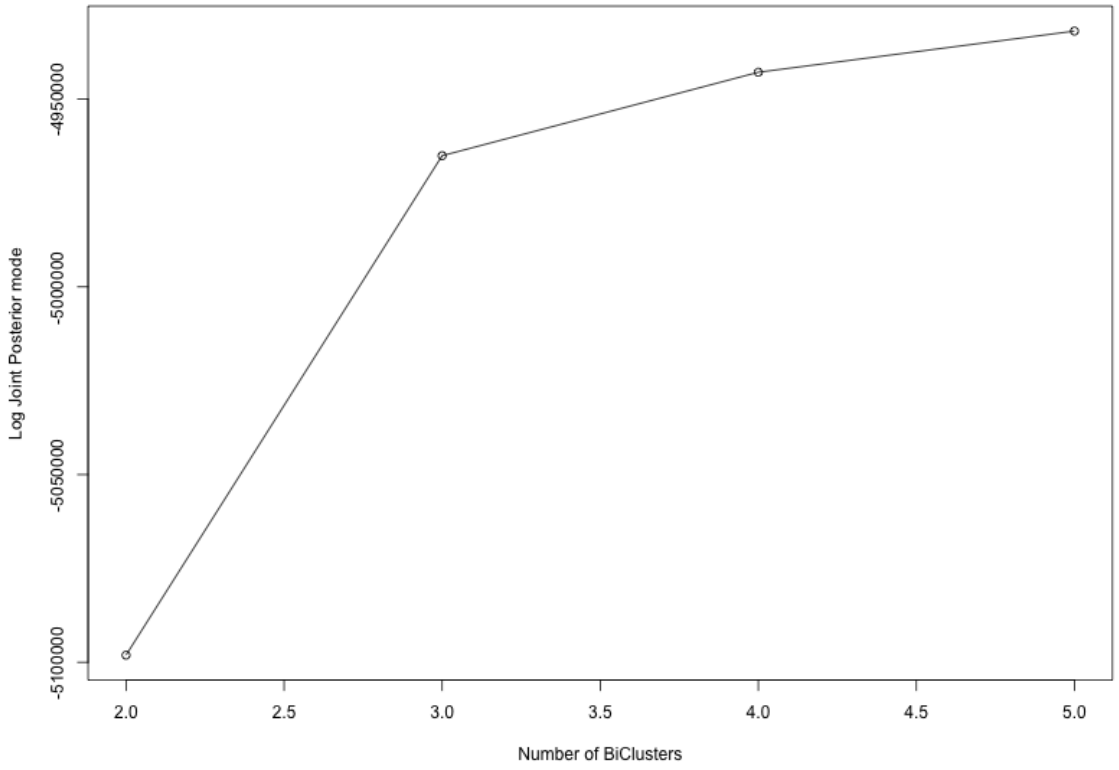

Figure 5.4: Comparison of joint posterior modes at 2, 3, 4, and 5 BiClusters.

The number of clusters was chosen to be 5 for the SNP full data set. The trace plot and ACF plot for 5 clusters are presented in Figure 5.5 and 5.6

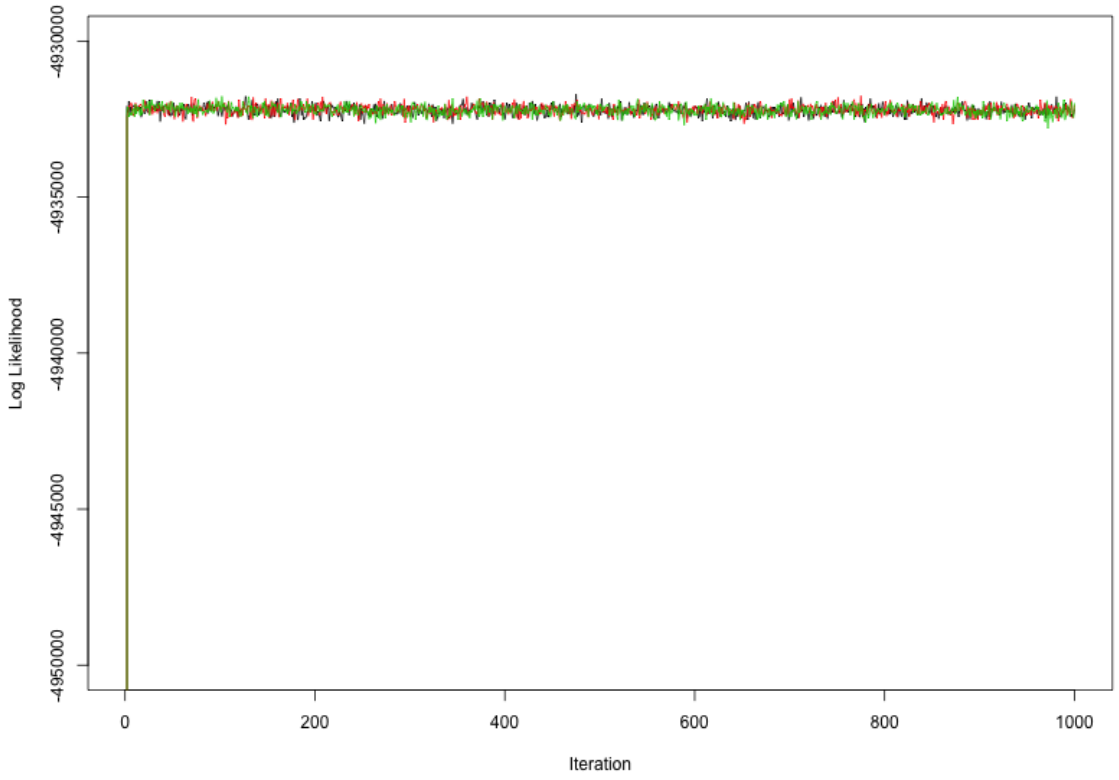

Figure 5.5: The trace plot of log-likelihood in 3 independent chains.

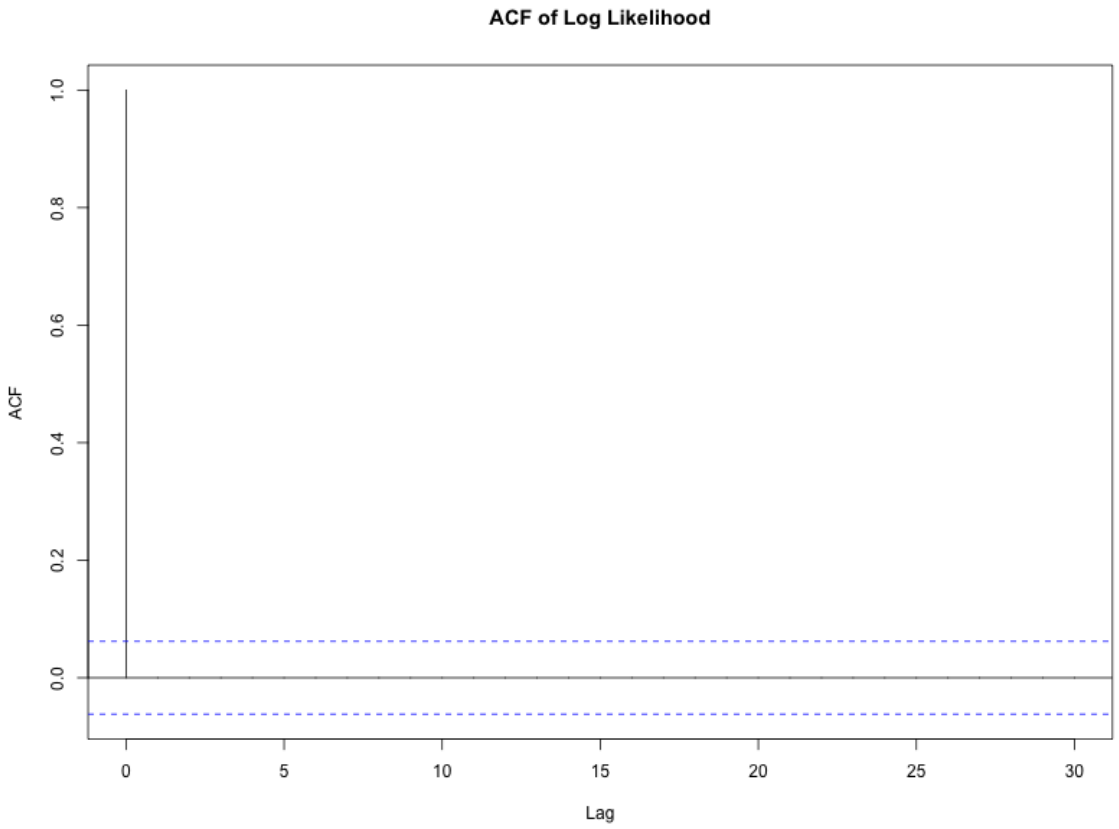

Figure 5.6: The ACF plot for Log-Likelihood

If we start with 2 clusters, the algorithm will converge at 2 clusters, as in Figure 5.7. Although our model allows the number of clusters to change dynamically, it is often trapped in a local mode because of the high energy barrier to jump to a higher number of clusters. From the plot, we can see that 2,847 SNPs differentiate individuals with African ancestry with all other individuals.

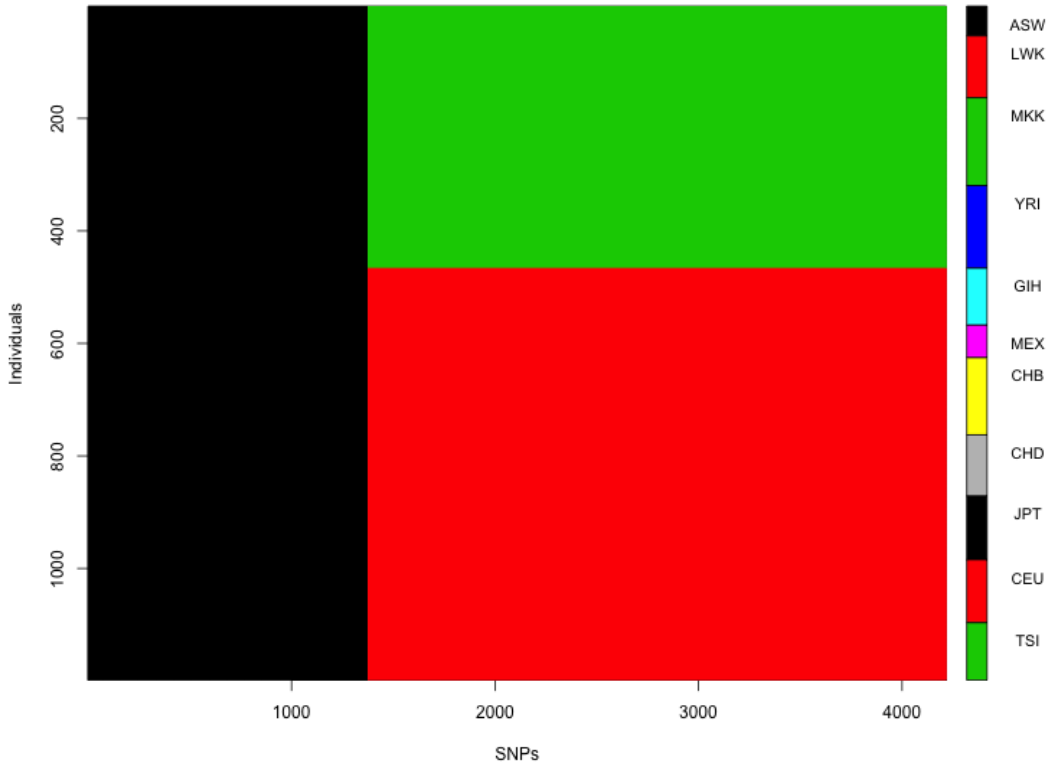

Figure 5.7: 2 BiClusters identified by starting with 2 clusters

The two biclusters identified are: Group 1: ASW LWK MKK YRI; Group 2: CHB CHD JPT CEU GIH MEX TSI.

If we start with 3 clusters, it will converge to 3 clusters at a local mode. A new cluster emerges from the second cluster of the previous analysis: individuals with East Asian ancestry are separated from individuals with Indian, Mexican and European ancestries. All individuals with African ancestries are in the same cluster as in the previous analysis. This can be intepreted as that Indian, Mexican, European and East Asian are closer in genetics compared to African ancestry. In addition, Indian,

Mexican and European are closer to each other than those of East Asian ancestry. The results are presented in Figure 5.8

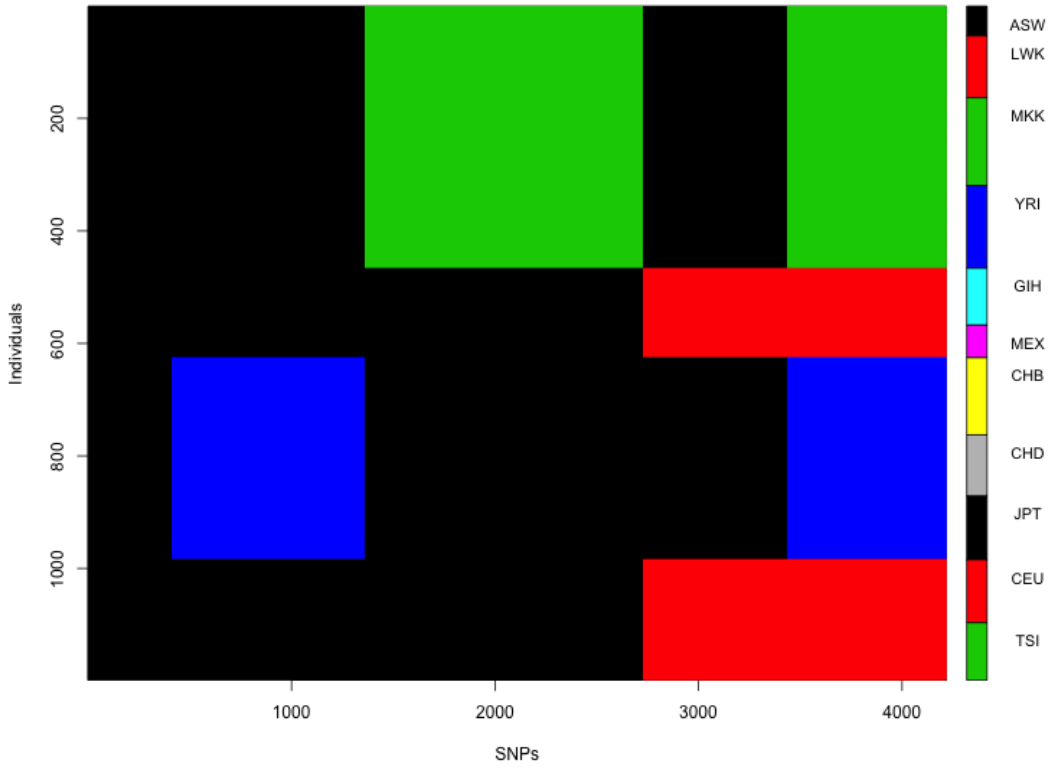

Figure 5.8: 3 BiClusters identified by starting with 3 clusters

The three biclusters identified are: Group 1: ASW LWK MKK YRI; Group 2: CHB CHD JPT; Group 3: CEU GIH MEX TSI.

We then start with 4 clusters, it will stay at 4 clusters because of the local mode. A new cluster emerges as the Indian group, while Mexican, European ancestry individuals are still in the same cluster. The results are shown in Figure 5.9

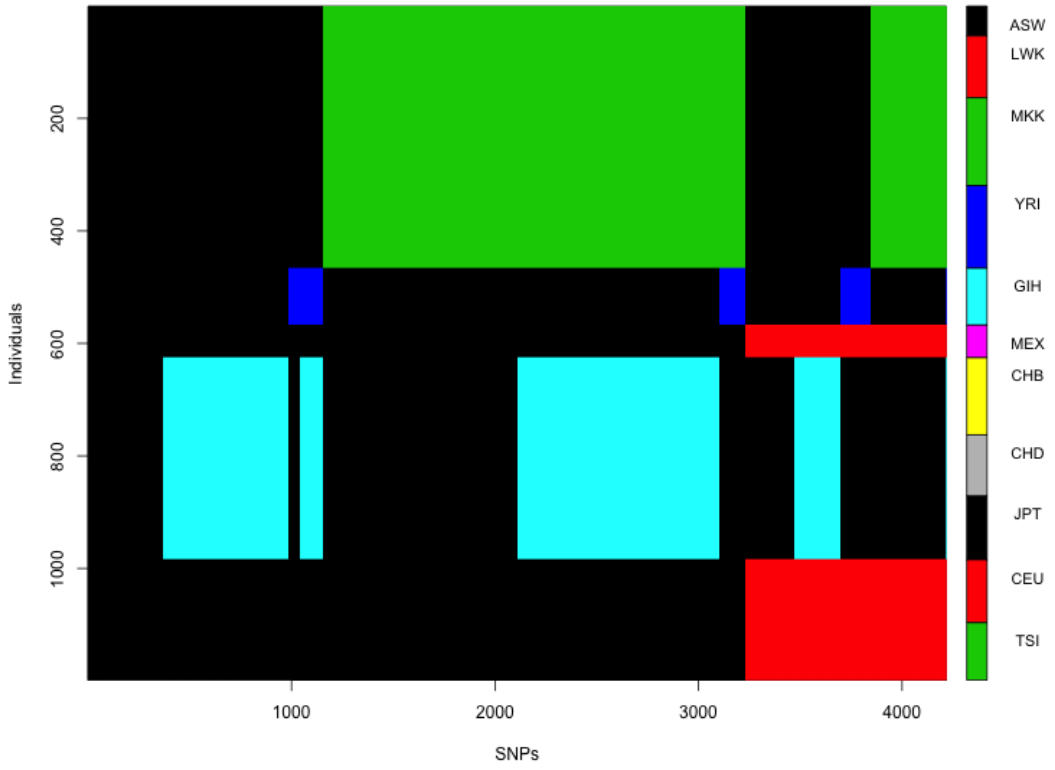

Figure 5.9: 4 BiClusters identified by starting with 4 clusters

The four biclusters identified are: Group 1: ASW LWK MKK YRI; Group 2: CHB CHD JPT; Group 3: GIH; Group 4: CEU MEX TSI.

If we start with 5 clusters or higher, the algorithm will converge to 5 as the total number of clusters. As seen in Figure 5.10, a new cluster emerges as the Mexican group, while CEU, TSI individuals are still in the same cluster. There are 3 individuals from the Mexican group that were assigned to the European ancestry group which includes CEU and TSI individuals. This is potentially due to the admixture nature of Mexican ancestry. These 3 individuals may inherit more genetic information from

European ancestry than from other sources. By comparing the joint posterior modes for 2, 3, 4 and 5 clusters, we chose 5 as the optimal number of clusters for the HapMap data set.

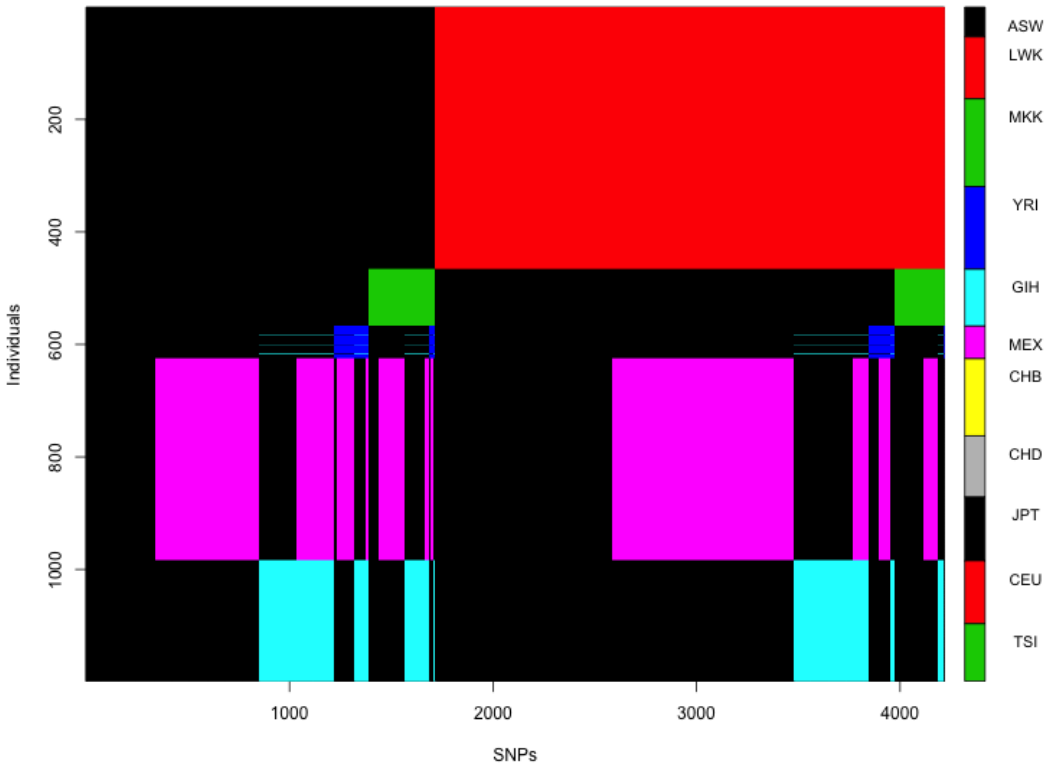

Figure 5.10: 5 BiClusters identified by starting with 5 clusters

The five biclusters identified are: Group 1: ASW LWK MKK YRI; Group 2: CHB CHD JPT; Group 3: GIH; Group 4: MEX; Group 5: CEU TSI. Table 5.2 lists the genetic similarity in terms of SNPs between these 5 population groups. 0 means the genotype frequencies of SNPs are from the same multinomial distribution. 343 SNPs are shared across all 5 populations. Table 5.3 lists the number of SNPs that are

unique to each of the population group. In other words, the number of SNPs that can distinguish the specific population group from other population groups. As we can see, African ancestry group has the most unique SNPs (891), followed by East Asian group (495), then the Northwest European group (185). This is consistent with the human migration theory, which assumes all humans originated in Africa, and then migrated to other parts of the world.

The number of SNPs shared by any two population groups are caculated and plotted in Figure 5.11. The Mexican and North Indian groups have 15 and 58 unique SNPs respectively, which suggests these two might be admixture population groups. Further looking into those 2 groups, we see that Mexican shares 71.4% of the 4,217 SNPs with the Northwest European group, 47.8% with the East Asian group, and 35.8% with the African group; the North Indian group shares 66.9% with the Northwest European group, 43.5% with the East Asian group and 32.6% with the African group; Between the North Indian and Mexican groups, there is a 80.2% similarity in the 4,217 SNPs. A possible cause might be that they share the similar ancestries as admixture population groups.

Table 5.2: Genetic differences among 5 discovered population groups

African	N Indian	Mexican	NW European	E Asian	Num of SNPs
0	0	0	0	0	343
0	0	0	0	1	495
0	0	0	1	0	185
0	0	0	1	1	184
0	0	1	0	0	15
0	0	1	0	1	79
0	0	1	1	0	56
0	0	1	1	1	20
0	1	0	0	0	58
0	1	0	0	1	124
0	1	0	1	0	92
0	1	0	1	1	27
0	1	1	0	0	7
0	1	1	0	1	15
0	1	1	1	0	5
1	0	0	0	0	891
1	0	0	0	1	912
1	0	0	1	0	286
1	0	0	1	1	84
1	0	1	0	0	36
1	0	1	0	1	49
1	0	1	1	0	22
1	1	0	0	0	127
1	1	0	0	1	62
1	1	0	1	0	33
1	1	1	0	0	10
1	1	1	1	1	0
					Total: 4217

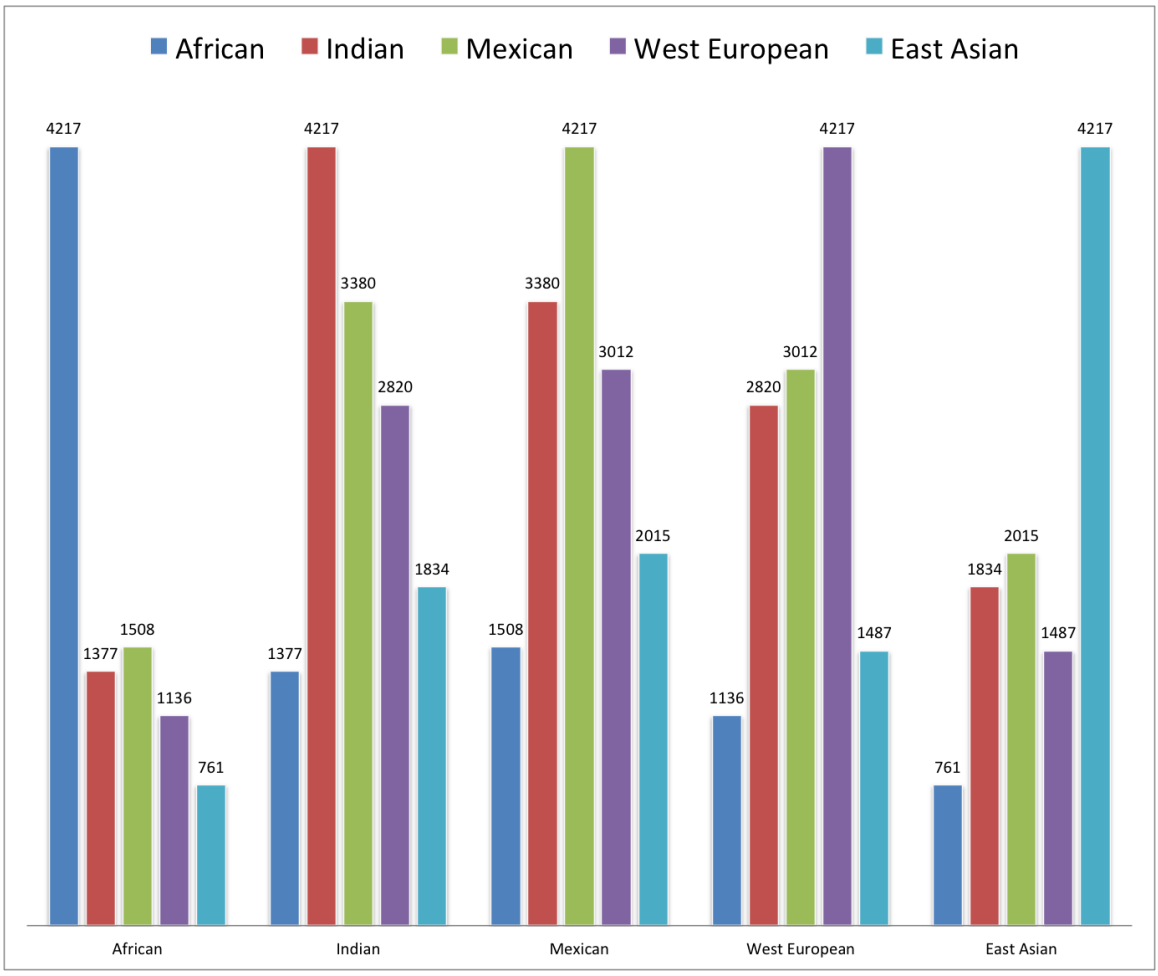

Figure 5.11: SNPs shared between populations groups.

Table 5.3: Number of Unique SNPs among 5 discovered population groups.

Population Group	Number of Unique SNPs
African	891
East Asian	495
Mexican	15
North Indian	58
Northwest European	185

Disease SNP linkage

Using the data from the published GWAS loci for human diseases and traits catalog (Hindorff et al. (2009), Hindorff et al. (2011)), we matched our 4,217 SNPs and found 27 SNPs that are known to be linked to diseases or traits. Table 5.4 lists the matched SNPs and their estimated allele frequencies in different population groups. The higher the allele frequency for the strongest risk allele, the more likely the population group is to be susceptible to the disease.

From the GWAS catalog, we know SNP $rs2071748$ is linked to obesity-related traits in human. The strongest SNP risk allele is **A**, in our estimate African, North Indian and East Asian group all have this **A** allele frequency at 0.53, while Mexican and Northwest European groups are at $0.22, 0.42$ respectively. We can speculate that African, North Indian and East Asian population groups have different susceptibility to this specific SNP linked traits due to their different risk allele frequencies from other groups. Another example is from SNP $rs12933233$, which is linked to Alzheimer's disease. Its strongest SNP risk allele is unknown. However, from our estimate we know that East Asian (0.65) has a higher allele frequency for allele **{A}** at this locus than any other population groups (African: 0.47; North Indian: 0.37; Mexican: 0.37; Northwest European: 0.37). East Asian group may have a different susceptibility to this disease than all other populations, depending on what the strongest SNP risk allele is. SNP $rs4964469$ is linked to Parkinson's disease according to the GWAS catalog, which has the strongest risk allele as **A**. Our estimate (African: 0.61; North Indian: 0.39; Mexican: 0.39; Northwest European: 0.39; East Asian: 0.39) suggests that African alone have different susceptibility to this linkage. The GWAS catalog

Table 5.4: Public GWAS disease catalog matched SNPs and their estimated allele frequencies in different population groups. For unknown Strongest SNP-Risk allele, the estimate is based on the minor allele at given locus.

SNP Name	Disease/Trait	S. Risk	African	N Indian	Mexican	NW European	E Asian
rs6601327	Multiple myeloma	G	0.49	0.49	0.49	0.36	0.6
rs2071748	Obesity-related traits	A	0.53	0.53	0.22	0.42	0.53
rs907121	Weight	C	0.44	0.59	0.44	0.65	0.44
rs727428	Sex hormone-binding globulin levels	T	0.53	0.53	0.53	0.44	0.53
rs10113903	IgG glycosylation	C	0.58	0.36	0.36	0.36	0.61
rs27855	Height	A	0.81	0.4	0.4	0.4	0.21
rs10906189	QT interval	A	0.55	0.55	0.55	0.55	0.42
rs11013962	Common traits (Other)	?	0.68	0.48	0.48	0.48	0.32
rs12933233	Alzheimer's disease	?	0.47	0.37	0.37	0.37	0.65
rs9951150	Autism spectrum disorder, etc	A	0.38	0.52	0.52	0.52	0.64
rs6030171	IgG glycosylation	C	0.51	0.34	0.34	0.34	0.64
rs1204798	Dental caries	?	0.2	0.72	0.72	0.72	0.72
rs6478241	Migraine	A	0.75	0.35	0.35	0.35	0.35
rs531676	Metabolic syndrome	?	0.38	0.61	0.61	0.5	0.61
rs3011225	Amyotrophic lateral sclerosis	G	0.78	0.32	0.32	0.32	0.32
rs3129882	Parkinson's disease	G	0.45	0.45	0.45	0.45	0.64
rs11977526	Insulin-like growth factors	A	0.4	0.4	0.4	0.4	0.82
rs2281636	Obesity-related traits	A	0.57	0.37	0.37	0.37	0.6
rs1292053	Inflammatory bowel disease	G	0.53	0.42	0.42	0.42	0.56
rs293428	Sex hormone-binding globulin levels	A	0.44	0.74	0.4	0.74	0.39
rs514024	Eating disorders	A	0.57	0.57	0.57	0.57	0.28
rs1354774	Prostate-specific antigen levels	G	0.88	0.41	0.41	0.41	0.21
rs1473247	Mean platelet volume	C	0.68	0.4	0.4	0.24	0.4
rs4537545	C-reactive protein	T	0.69	0.4	0.4	0.4	0.4
rs483610	Obesity-related traits	G	0.74	0.35	0.35	0.35	0.35
rs7837791	Refractive error	T	0.61	0.22	0.74	0.51	0.51
rs4964469	Parkinson's disease	A	0.61	0.39	0.39	0.39	0.39

has been growing fast with contributions from association study and we will have more data to investigate how various disease-linked SNPs interplay with race-specific SNPs as it grows.

2,000 SNPs

We randomly draw 2,000 SNPs from the 4,217 SNPs and use this subset as input, our algorithm can still separate the 1,198 individuals into 5 major population groups. The estimated bicluster structure is shown in Figure 5.12. The population specific SNPs pattern looks similar to the one we obtained using all 4,217 SNPs. A detailed comparison of similarity patterns is presented in Table 5.5.

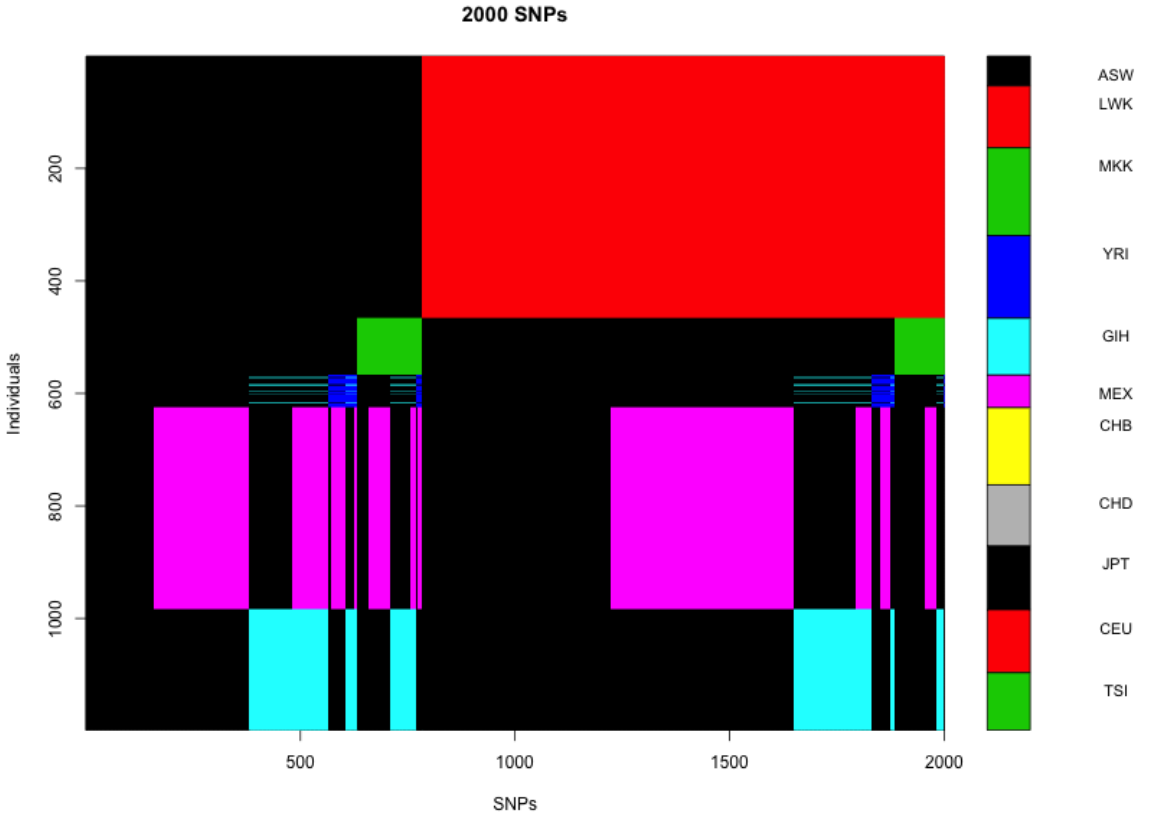

Figure 5.12: Population structure estimated using 2,000 SNPs

1,000 SNPs

If we only draw 1,000 SNPs and repeat our alogrithm, we can also get the same clusters. Table 5.5 lists the percentage of population specific SNPs from experiments with different number of SNPs used. We can see the percentage is very consistent between using 4,217 SNPs and 2,000 SNPs. It is no longer consistent with 1,000 SNPs.

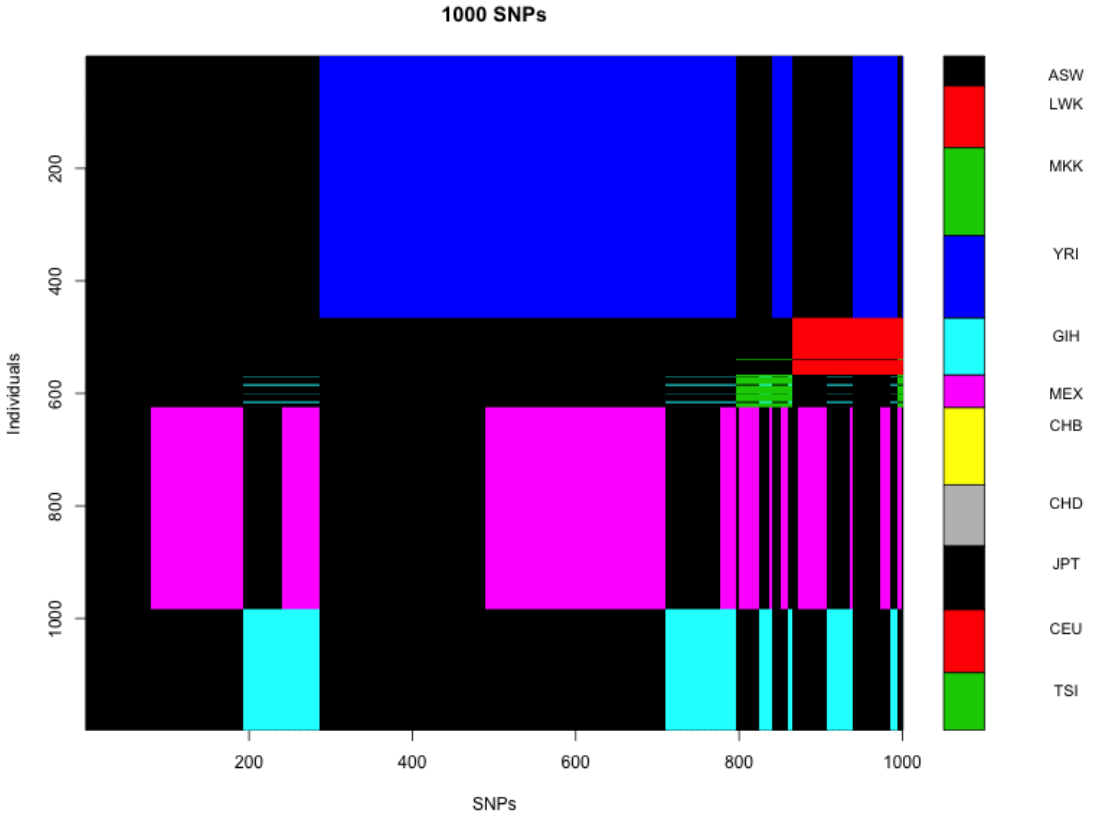

Figure 5.13: Population structure estimated using 1,000 SNPs

Table 5.5: Estimated population specific SNPs using different number of SNPs

Num of SNPs	African	N Indian	Mexican	NW European
4,217	891 (21.1%)	58 (1.38%)	15 (0.36%)	185 (4.39%)
2,000	440 (22%)	26 (1.3%)	8 (0.4%)	100 (5%)
1,000	7 (0.7%)	5 (0.5%)	204 (20.4%)	47 (4.7%)

500 SNPs

We keep decreasing the number of SNPs in our experiment. When the number drops to 500, it can only form 3 clusters: (African, East Asian, (Mexican, North

Indian, Northwest European)), as plotted in Figure 5.14. We tried multiple experiements and the results were similar.

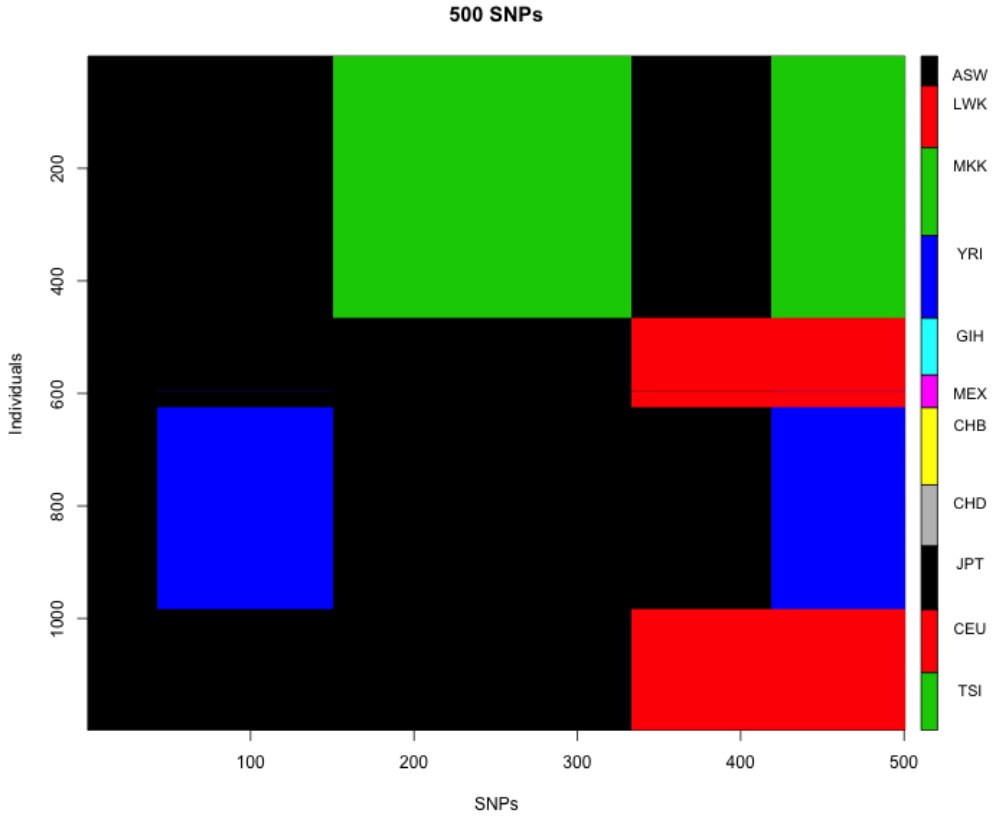

Figure 5.14: Population structure estimated using 500 SNPs

200 SNPs

When we drop the number of SNPs to 200. The algorithm keeps adding new clusters by separating individuals from Indian and Mexican populations into new clusters. We tried 5 independent random draws and the results are the same. We can see that 1,000 SNPs is probably the lower bound for the number of SNPs required for separating individuals into 5 major population groups. We can speculate that

with more independent SNPs added, more meaningful clusters might form using our Bayesian Biclustering algorithm.

5.4.2 Variable Selection for STRUCTURE

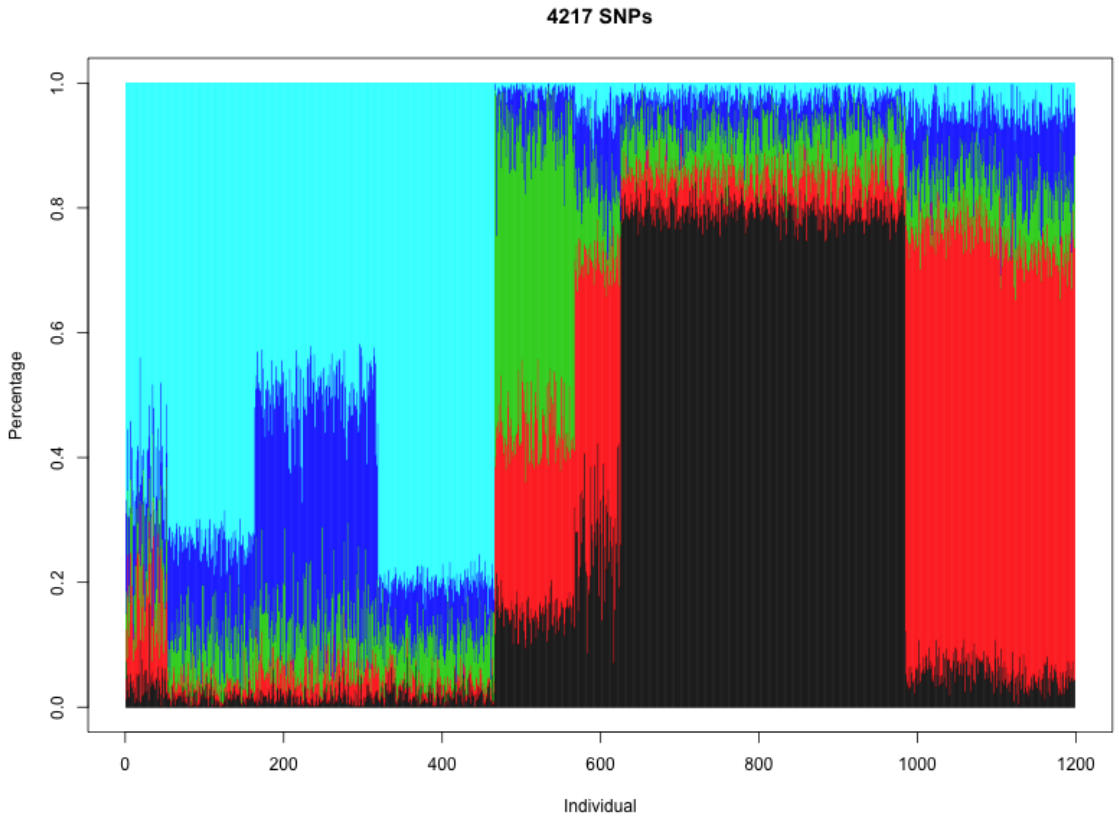

Figure 5.15: STRUCTURE analysis with 4,217 SNPs

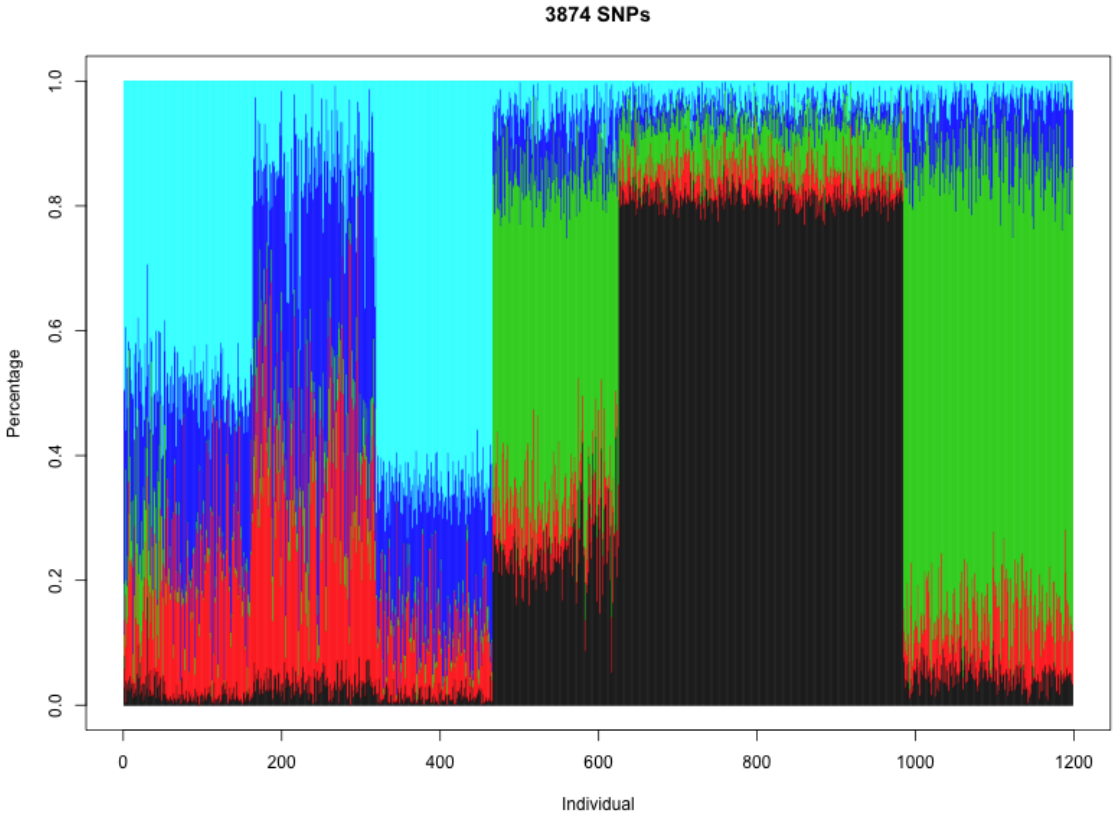

Figure 5.16: STRUCTURE analysis with 3,874 SNPs

As we discussed before, our Bayesian Biclustering model can be used as a variable selection step. We use all 4,217 SNPs for 1,198 individuals as input to STRUCTURE and set the number of ancestries to be 5. The estimated percentage of ancestries for each individual is plotted in Figure 5.15. Every vertical line represents the ancestral composition of an individual, with the length of the colored segements proportional to the percentage of corresponding ancestries. After removing the 343 commonly shared SNPs identified by our algorithm, we performed the analysis again and the estiamted percentages are presented in Figure 5.16.

Comparing the 2 plots, we can see that removing the 343 SNPs (about 8% of the total SNPs used) will have little impact on the percentage estimate for East Asian populations, while there is huge change in Indian, Mexican and Northwest European populations. The *green* ancestry in Indian,Mexican and Northwest European population becomes dominant. Mexican and Indian populations becomes even more similar to each other after the removal according to STRUCTURE. In the meantime, the removal caused another ancestry percentage become more evident in African populations(the red segments). The dramatical change in percentage of ancestries are subject to further investigation.

5.4.3 Discussion

In our analysis, we did not use any of the population information and started everything from an unsupervised standing point. However, the biclustering results show a high consistency between real geographical population groups and our predicted groups. This reveals the fact that human genome stores valuable and powerful information to differentiate people. In our analysis, we could not separate some ethinic groups into distinctive clusters. This might be caused by the using of only 4,217 SNPs due to limitations of computing power.

Our choice of 10^6 as the correlation threshold for choosing Tag SNPs is also arbitary. Given more computing power, we can relax this constraint by including more Tag SNPs. In the meantime, the categorical biclustering model can be further improved by allowing correlations between SNPs, including genetic proximity between SNPs on chromosomes.

Bibliography

David M Altshuler, Richard A Gibbs, Leena Peltonen, Emmanouil Dermitzakis, Stephen F Schaffner, Fuli Yu, Penelope E Bonnen, PI De Bakker, Panos Deloukas, Stacey B Gabriel, et al. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467(7311):52–58, 2010.

Jill S Barnholtz-Sloan, Brian McEvoy, Mark D Shriver, and Timothy R Rebbeck. Ancestry estimation and correction for population stratification in molecular epidemiologic association studies. *Cancer Epidemiology Biomarkers & Prevention*, 17 (3):471–477, 2008.

Marc Bauchet, Brian McEvoy, Laurel N Pearson, Ellen E Quillen, Tamara Sarkisian, Kristine Hovhannesyan, Ranjan Deka, Daniel G Bradley, and Mark D Shriver. Measuring european population stratification with microarray genotype data. *The American Journal of Human Genetics*, 80(5):948–956, 2007.

Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. In *Proceedings of the sixth annual international conference on Computational biology*, pages 49–57. ACM, 2002.

Sven Bergmann, Jan Ihmels, and Naama Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review E*, 67(3):031902, 2003.

Paul R Burton, David G Clayton, Lon R Cardon, Nick Craddock, Panos Deloukas, Audrey Duncanson, Dominic P Kwiatkowski, Mark I McCarthy, Willem H Ouwehand, Nilesh J Samani, et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, 2007.

Neil Caporaso, Nathaniel Rothman, and Sholom Wacholder. Case-control studies of common alleles and environmental factors. *JNCI Monographs*, 1999(26):25–30, 1999.

Lon R Cardon and Lyle J Palmer. Population stratification and spurious allelic association. *The Lancet*, 361(9357):598–604, 2003.

Yizong Cheng and George M Church. Biclustering of expression data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology*, volume 8, pages 93–103, 2000.

Shweta Choudhry, Natasha E Coyle, Hua Tang, Keyan Salari, Denise Lind, Suzanne L Clark, Hui-Ju Tsai, Mariam Naqvi, Angie Phong, Ngim Ung, et al. Population stratification confounds genetic association studies among latinos. *Human genetics*, 118(5):652–664, 2006.

C Clark Cockerham and BS Weir. Estimation of gene flow from f-statistics. *Evolution*, pages 855–863, 1993.

Nick Craddock, Matthew E Hurles, Niall Cardin, Richard D Pearson, Vincent Plagnol, Samuel Robson, Damjan Vukcevic, Chris Barnes, Donald F Conrad, Eleni Giannoulatou, et al. Genome-wide association study of cnvs in 16,000 cases of eight common diseases and 3,000 shared controls. *Nature*, 464(7289):713–720, 2010.

Michiel JL de Hoon, Seiya Imoto, John Nolan, and Satoru Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004.

B Devlin, Silviu-Alin Bacanu, and Kathryn Roeder. Genomic control to the extreme. *Nature genetics*, 36(11):1129–1130, 2004.

Michael P Epstein, Andrew S Allen, and Glen A Satten. A simple and improved correction for population stratification in case-control studies. *The American Journal of Human Genetics*, 80(5):921–930, 2007.

Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587, 2003.

Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular Ecology Notes*, 7(4):574–578, 2007.

Matthew L Freedman, David Reich, Kathryn L Penney, Gavin J McDonald, Andre A Mignault, Nick Patterson, Stacey B Gabriel, Eric J Topol, Jordan W Smoller, Carlos N Pato, et al. Assessing the impact of population stratification on genetic association studies. *Nature genetics*, 36(4):388–393, 2004.

Gad Getz, Erel Levine, and Eytan Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22): 12079–12084, 2000.

Katrina AB Goddard, Penelope J Hopkins, Jeff M Hall, and John S Witte. Linkage disequilibrium and allele-frequency distributions for 114 single-nucleotide polymorphisms in five populations. *The American Journal of Human Genetics*, 66(1): 216–234, 2000.

Jiajun Gu and Jun S Liu. Bayesian biclustering of gene expression data. *BMC genomics*, 9(Suppl 1):S4, 2008.

Eran Halperin, Gad Kimmel, and Ron Shamir. Tag snp selection in genotype data for maximizing snp prediction accuracy. *Bioinformatics*, 21(suppl 1):i195–i203, 2005.

John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129, 1972.

Lucia A Hindorff, Praveen Sethupathy, Heather A Junkins, Erin M Ramos, Jayashri P Mehta, Francis S Collins, and Teri A Manolio. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences*, 106(23):9362–9367, 2009.

Lucia A Hindorff, Heather A Junkins, JP Mehta, TA Manolio, et al. A catalog of published genome-wide association studies. 2011.

Melissa J Hubisz, Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inferring weak population structure with the assistance of sample group information. *Molecular ecology resources*, 9(5):1322–1332, 2009.

David J Hunter, Peter Kraft, Kevin B Jacobs, David G Cox, Meredith Yeager, Susan E Hankinson, Sholom Wacholder, Zhaoming Wang, Robert Welch, Amy Hutchinson, et al. A genome-wide association study identifies alleles in fgfr2 associated with risk of sporadic postmenopausal breast cancer. *Nature genetics*, 39(7): 870–874, 2007.

P Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques regions voisines. *Bulletin de la Sociate Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

J Edward Jackson. *A user's guide to principal components*, volume 244. Wiley-Interscience, 2005.

Andrew D Johnson and Christopher J O'Donnell. An open access database of genome-wide association results. *BMC medical genetics*, 10(1):6, 2009.

Thibaut Jombart, Sébastien Devillard, and François Balloux. Discriminant analysis of principal components: a new method for the analysis of genetically structured populations. *BMC genetics*, 11(1):94, 2010.

Sebastian Kaiser and Friedrich Leisch. A toolbox for bicluster analysis in r. 2008.

Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13 (4):703–716, 2003.

William C Knowler, RC Williams, DJ Pettitt, and A Gm Steinberg. Gm3; 5, 13, 14 and type 2 diabetes mellitus: an association in american indians with genetic admixture. *American journal of human genetics*, 43(4):520, 1988.

Laura Lazzeroni and Art Owen. Plaid models for gene expression data. *Statistica Sinica*, 12(1):61–86, 2002.

Qizhai Li and KAI Yu. Improved correction for population stratification in genome-wide association studies by identifying hidden population structures. *Genetic epidemiology*, 32(3):215–226, 2008.

Jun S Liu. *Monte Carlo strategies in scientific computing*. springer, 2008.

Jun S Liu, Faming Liang, and Wing Hung Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(1):24–45, 2004.

N Malats and F Calafell. Basic glossary on genetic epidemiology. *Journal of epidemiology and community health*, 57(7):480–482, 2003.

Xianyun Mao, Abigail W Bigham, Rui Mei, Gerardo Gutierrez, Ken M Weiss, Tom D Brutsaert, Fabiola Leon-Velarde, Lorna G Moore, Enrique Vargas, Paul M McKeigue, et al. A genomewide admixture mapping panel for hispanic/latino populations. *The American Journal of Human Genetics*, 80(6):1171–1178, 2007.

Boris Grigor'evič Mirkin. *Mathematical classification and clustering*, volume 11. Kluwer Academic Pub, 1996.

TM Murali and Simon Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proc. Pacific Symp. Biocomputing*, volume 3, pages 77–88, 2003.

United Nations. World population prospects: The 2012 revision, key findings and advance tables. *Working Paper*, (No. ESA/P/WP. 227), 2013.

Esteban J Parra, Amy Marcini, Joshua Akey, Jeremy Martinson, Mark A Batzer, Richard Cooper, Terrence Forrester, David B Allison, Ranjan Deka, Robert E Ferrell, et al. Estimating african american admixture proportions by use of population-specific alleles. *The American Journal of Human Genetics*, 63(6):1839–1851, 1998.

A Perez-Lezaun, F Calafell, E Mateu, D Comas, E Bosch, and J Bertranpetit. Allele frequencies for 20 microsatellites in a worldwide population survey. *Human heredity*, 47(4):189–196, 1997.

Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, 2006.

Alkes L Price, Noah A Zaitlen, David Reich, and Nick Patterson. New approaches to population stratification in genome-wide association studies. *Nature Reviews Genetics*, 11(7):459–463, 2010.

Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

David E Reich, Michele Cargill, Stacey Bolk, James Ireland, Pardis C Sabeti, Daniel J Richter, Thomas Lavery, Rose Kouyoumjian, Shelli F Farhadian, Ryk Ward, et al. Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204, 2001.

Keyan Salari, Shweta Choudhry, Hua Tang, Mariam Naqvi, Denise Lind, Pedro C Avila, Natasha E Coyle, Ngim Ung, Sylvette Nazario, Jesus Casal, et al. Genetic admixture and asthma-related phenotypes in mexican american and puerto rican asthmatics. *Genetic epidemiology*, 29(1):76–86, 2005.

Erika Salvi, Alessandro Orro, Guia Guffanti, Sara Lupoli, Federica Torri, Cristina Barlassina, Steven Potkin, Daniele Cusi, Fabio Macciardi, and Luciano Milanesi. Population stratification analysis in genome-wide association studies. In *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, pages 177–196. Springer, 2011.

Thomas D Schneider and R Michael Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20):6097–6100, 1990.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6 (2):461–464, 1978.

Eran Segal, Ben Taskar, Audrey Gasch, Nir Friedman, and Daphne Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17(suppl 1):S243–S252, 2001.

Michael F Seldin and Alkes L Price. Application of ancestry informative markers to association studies in european americans. *PLoS genetics*, 4(1):e5, 2008.

David Serre, Alexandre Montpetit, Guillaume Paré, James C Engert, Salim Yusuf, Bernard Keavney, Thomas J Hudson, and Sonia Anand. Correction of population stratification in large multi-ethnic association studies. *PLoS One*, 3(1):e1382, 2008.

Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl 1):S136–S144, 2002.

Amos Tanay, Roded Sharan, and Ron Shamir. Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 9:26–1, 2005.

Gudmundur A Thorisson, Albert V Smith, Lalitha Krishnan, and Lincoln D Stein. The international hapmap project web site. *Genome research*, 15(11):1592–1593, 2005.

Chao Tian, David A Hinds, Russell Shigeta, Rick Kittles, Dennis G Ballinger, and Michael F Seldin. A genomewide single-nucleotide–polymorphism panel with high ancestry information for african american admixture mapping. *The American Journal of Human Genetics*, 79(4):640–649, 2006.

Chao Tian, Robert M Plenge, Michael Ransom, Annette Lee, Pablo Villoslada, Carlo Selmi, Lars Klareskog, Ann E Pulver, Lihong Qi, Peter K Gregersen, et al. Analysis and application of european genetic substructure using 300 k snp information. *PLoS genetics*, 4(1):e4, 2008.

Haixun Wang, Wei Wang, Jiong Yang, and Philip S Yu. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 394–405. ACM, 2002.

Bruce S Weir and C Clark Cockerham. Estimating f-statistics for the analysis of population structure. *evolution*, pages 1358–1370, 1984.

Jiong Yang, Wei Wang, Haixun Wang, and Philip Yu. δ-clusters: Capturing subspace correlation in a large data set. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 517–528. IEEE, 2002.

Jiong Yang, Haixun Wang, Wei Wang, and Philip Yu. Enhanced biclustering on expression data. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*, pages 321–327. IEEE, 2003.

Meredith Yeager, Nick Orr, Richard B Hayes, Kevin B Jacobs, Peter Kraft, Sholom Wacholder, Mark J Minichiello, Paul Fearnhead, Kai Yu, Nilanjan Chatterjee, et al. Genome-wide association study of prostate cancer identifies a second risk locus at 8q24. *Nature genetics*, 39(5):645–649, 2007.